INFORMATION THEORETIC CAUSALITY MEASURES FOR PARAMETER ESTIMATION AND SYSTEM IDENTIFICATION

A Dissertation
Presented to
The Academic Faculty

By

Jared Elinger

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Mechanical Engineering

Georgia Institute of Technology

December 2020

INFORMATION THEORETIC CAUSALITY MEASURES FOR PARAMETER ESTIMATION AND SYSTEM IDENTIFICATION

Approved by:

Dr. Jonathan Rogers, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Aldo Ferri
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Frank Hammond
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Michael Leamy
School of Mechanical Engineering
Georgia Institute of Technology

Dr. Panagiotis Tsiotras
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: August 26, 2020

To my friends and family.

ACKNOWLEDGEMENTS

The progression towards my PhD and the results included herein are by no means an individual effort; I'd like to thank the variety of people who have helped greatly with my time as a PhD student at Georgia Tech. I'd like to begin by thanking my advisor, Dr. Jonathan Rogers for his outstanding support, mentorship and time given as I researched under him. His willingness to provide flexibility on topics considered and avenues taken while still providing thoughtful guidance and help made the entire process very rewarding. I would also like to thank the professors whose classes I took while at Georgia Tech; they provided excellent insight that helped with both research as well as expanded my horizons to new areas that will certainly assist in my career beyond this program. Finally, I'd like to thank the members of my committee for their continued time and insights to this work.

I would also like to thank the various friends I've made at Georgia Tech who have assisted with classwork, research, and quals preparation all while making my time more enjoyable. I'd like to thank all the members of the iREAL lab from through out my time in it including Dr. Jonathan Warner and Andrew Leonard as well as Brian, Joey, Kevin, Evan, Dakota, Geordan, Sam and Adam. You all helped make my time both entertaining and stimulating. You were all also indispensible in helping iREAL et. al. to school-wide glory and returning graduate students to their proper throne. I'd also like to thank Vansh for his assistance as an undergraduate researcher in the lab for helping develop the instrumentation of the experimental testbed used. I'd also like to thank my friend Michael LiBretto for his available at any time help along with the long hours spent working on course work through our first couple of years. I'd also like to thank my Uncle Pat and cousins Edward, Phil and Bobby for their invaluable help with constructing my experimental setup while Georgia Tech was closed.

I'd also like to thank my friends and family for their encouragement and support. Pursuing a PhD is certainly stressful at times, and it wouldn't have been possible without you

all. I'd like to give a special shout out to my parents for their unconditional love and support in helping me pursue my goal of completing my PhD at Georgia Tech.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Constructing a model for a dynamic system from observed data is a complicated yet common problem for many engineered systems. This task, known as system identification, is a necessary step in many fields of engineering as it is often used for system modeling, simulation and control design. Inaccurate system models can lead to poor simulation results, which will lead to poor real world performance. While linear systems have a set of developed identification techniques, methods for nonlinear systems are not as generalizable or robust. Parameter estimation is a subset of system identification where a model structure is selected (either through first principles creating a grey box model or if a pre-prescribed structure is used for a black box model) with a set of parameters corresponding to the model needing to be optimized. In the case of linear systems, the solution to the parameter estimation problem is a closed-form least squares solution; however, parameters of nonlinear systems must be solved for numerically, which is subject to well-known issues of the solution converging to a local extrema. Maximum Likelihood Estimation (MLE) is commonly used to optimize the nonlinear system parameter set by minimizing the least-squares error between the actual data and the candidate optimized model. This optimization can often converge to local extrema, especially in the case noise, exogenous disturbance, or when a relatively small number of data points is available when compared to the dimension of the optimization problem. A problem known as overfitting can occur when a more complex model than needed is considered, as potentially multiple high accuracy unique model fits can be found over the available training data that generalizes poorly to unseen data as the optimized model no longer matches the generative dynamics. Methods to determine the optimal parameter set to be both accurate and predictive are critical to creation of a high fidelity model; these techniques are frequently referred to as covariate selection or feature selection techniques. The recently proposed Causation Entropy Matrix (CEM) allows for identification of causal information flow within a system. This is of immediate usefulness

to many system identification tasks when the exact or entire structure of the system is unknown and covariate selection is needed. The CEM provides a method for pre-optimization, data-based covariate selection to allow for reduction of the number of parameters included in the system optimization to improve MLE results. This work provides background on the Causation Entropy Matrix and its computation before providing multiple examples of application of the CEM to grey-box and black-box modeling problems. The effectiveness of the Causation Entropy Matrix is then compared to the current state of the art techniques of LASSO (least absolute shrinkage and selection operator) and elastic net. Next, a chapter is dedicated to the practical considerations needed for application of the CEM to real-world systems including but not limited to noise, unmodeled dynamics, and sampling rate. This work concludes with a study of the application of the CEM to data experimentally collected from a physical, nonlinear system. The ability of the CEM to accurately identify the underlying structure of the generative dynamics demonstrating the method is a promising technique for nonlinear system identification and covariate selection.

CHAPTER 1
INTRODUCTION AND BACKGROUND

1.1 Problem Motivation

Constructing a dynamic model from observed data can be a quite difficult yet necessary task for many complex engineered systems. This process, termed system identification, is ubiquitous in nearly all fields of engineering and often comprises a key step in modeling, simulation, and control system design. Inaccurate modeling of the physical system inevitably leads to incorrect predictions and, in the case of actively-controlled devices, poor performance in real-world settings. While a variety of well-known system ID tools have been developed for linear systems [1, 2, 3, 4], there is a continuing need for nonlinear system ID methods that are generalizable and robust. One subset of the system ID problem is parameter estimation, in which the structure of the system is known or assumed, but the values for system parameters are unknown. If the true underlying structure of the system is at least somewhat known (usually from physical first principles), the problem is known as grey-box modeling, while if the structure is entirely unknown, but a selected model of a pre-prescribed structure is being fit it is known as black-box modeling [5]. Parameter estimation for linear systems yields a closed-form least squares solution [6]. In the case of nonlinear systems, however, the problem must be solved numerically with the results subject to well-known issues of convergence to local minima [7].

The nonlinear parameter estimation problem begins with formulation of a nonlinear dynamic model containing a set of either uncertain or wholly unknown parameters. The most commonly-used technique for estimating nonlinear model parameters from observed data is Maximum Likelihood Estimation (MLE) or output error minimization [8, 9]. Given a set of time series data, the optimized model attempts to minimize the error between the model

and actual data in the least-squares sense, yielding the optimal system parameters. This numerical optimization process oftentimes converges to local minima, especially when data is subject to noise or exogenous disturbances [10]. The quality of the initial guess for the parameter set is well-known to have a significant effect on convergence to the global optimum when the optimization problem is non-convex [11].

Extended Kalman Filters, another form of MLE that is very similar to recursive least squares, are a commonly used technique for handling nonlinear system identification. However, extended Kalman Filters have multiple problems that can greatly restrict their success. The extended Kalman Filter is a modification of the Kalman Filter for linear systems that uses a first order approximation of nonlinearities to approximate the system; however, in highly nonlinear systems where complex nonlinearities can dominate, this approximation is not sufficient to yield high accuracy results. Higher order approximations have been proposed, but have been shown to suffer greatly in the presence of measurement noise. Additionally, Kalman Filters can struggle to converge as the number of parameters that need to be identified grows and/or if little apriori knowledge is known about the value of the parameters prior to initializing the filter [12, 13, 14].

One particular manifestation of the problem of converging to a local extrema during MLE is known as the problem of overfitting of data, which occurs when more than the minimal necessary number of parameters/functions is used to generate the model. This occurs when the optimization routine converges to a local extrema that uses more paramaters than are needed, which provides a very close fit over the training data but will have very poor generalizability to never before seen data [15]. This problem is particularly exacerbated in cases when the ratio of available training data points to the number of parameters/functions being fit is low [16]. Training data may be limited for a variety of reasons including but not limited to inaccessibility of the system or prohibitive cost considerations for running more systems tests. Therefore, a method that reduces the order of the optimization problem which removes corresponding potential local extrema and thus limits overfitting, without

making any assumptions on the numerical optimization technique used can provide vastly improved results. Methods that attempt to identify and remove wholly unnecessary or minimally necessary terms from the optimization problem are often referred to as covariate selection or feature identification techniques.

Recently, Kim *et al.* [17] proposed a novel information-theoretic technique to facilitate covariate selection for Maximum Likelihood Estimation by identifying the sparsity structure of the parameter set pre optimization. This work used an information theory measure of causal influence, called causation entropy [18], to measure the information transfer between sets of time series. By applying the causation entropy measure to time histories of the measured states, model parameters which should be removed from the parameter set (i.e., set to zero) can be immediately identified as the corresponding information flow is equal to zero. The goal behind the technique is to identify unnecessary parameter/potential state function pairs for pre-optimization removal. This will lead to improved optimization results as the reduced order problem will have fewer local extrema and a lower dimension search space. The intended modification of the work flow is shown in Figure 1.1. Notice that in Figure 1.1 additional steps of computing the CEM are added that will increase the computational cost of the problem, but will lead to a potentially improved model fit while using the identical numerical optimization technique over a reduced sized problem as compared to the originially formulated problem without system structure knowledge. This methodology is identical to existing covariate selection methods where the CEM block replaced with another covariate selection technique included in the box. Existing covariate selection techniques are discussed in the next section.

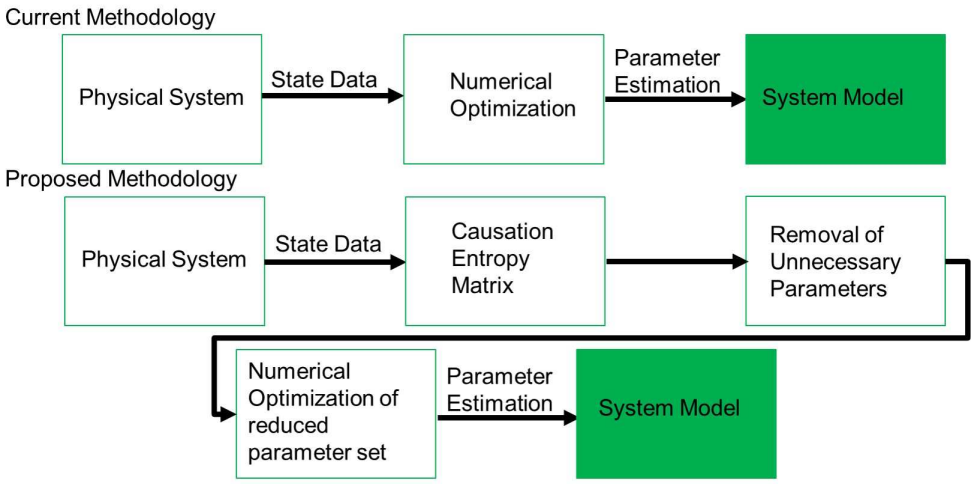

Figure 1.1: Visual representation of Causation Entropy Matrix methodology

Numerical results showed that, by applying the so-called Causation Entropy Matrix (CEM) prior to MLE, the resulting optimization solution was obtained faster and with increased the chances of convergence to the global optimum. While [17] established the overall methodology, the examples considered therein were relatively simple and restricted nonlinear systems that did not have direct engineering application. This work proposes a relaxed definition of the CEM that makes it applicable to most mechanical, nonlinear systems and fully explores the applicability and behavior of the technique.

1.1.1 Current Covariate Selection Techniques

Given the problems with numerical optimization used during MLE demonstrated above, the importance and benefits of apriori structure identification is well known and currently being explored and studied in many fields using various techniques and nomenclatures. In the field of machine learning the problem is frequently referred to as covariate selection and feature selection [19, 20]. Wrapper methods are one common class of covariate selection techniques that involve a search through potential subsets of features to maximize predictive accuracy [21, 22]. The methods work by selecting a subset of parameters, optimizing a model and then testing the model on a separate set of validation data. However, wrapper

techniques require a computationally expensive (and potentially intractable) comprehensive search along with significant amounts of independent data for training/validation sets that cover the entire phase space expected during model use. Another major class of techniques is referred to as filter methods. These methods use derived metrics (such as mutual information) or statistical tests (Anova, Chi-Square) to determine the correlation between the covariate and the output [23]. This feature by feature testing can struggle when having complex interrelations between features or the case of redundant features. The proposed CEM technique most closely fits or could be considered an improved filter method for covariate selection.

Recently, some regularization techniques (also often referred to as embedded methods) have emerged to improve upon regression and optimization results by sacrificing some bias to achieve a lower variance in the bias-variance tradeoff, which often comes in the form of a more sparse and simpler model [24]. Recently, the proposed methods of least absolute shrinkage and selection operator (LASSO) [25, 26] and elastic net [27] have provided promising results by using L_1 and L_2 penalties to encourage sparse results. These techniques simultaneously perform feature identification and parameter optimization.

Both the LASSO and elastic net methods are examples of shrinkage estimators that require the tuning of hyperparameters in order to successfully compute the models. Hyperparameter selection requires cross validation, which requires sufficient data to have independent training and validation sets for best results. Additionally, there are potential pitfalls of the LASSO and elastic net techniques. First, selection of the hyperparamters is a non-convex problem that can have multiple local extrema and thus converge to different models. This issue becomes particularly problematic when the hyperparameters are sampled for testing from a discrete set and thus the true minima may not even be included in those tested [28]. Additionally, it is not generally guaranteed that in the case of infinite data that LASSO will converge to the true model when presented with infinite training data; the chances of selecting the true model decreases even further in the case of limited data

with no guarantees on model performance. Particularly, LASSO can tend to select multiple closely related predictors when both are not necessary [29]. Elastic net solves some stability concerns of the LASSO algorithm and guarantees the convexity of the problem (though not the hyperparamter selection problem), it selects a less sparse model and still suffers from the same aforementioned issues [27]. Finally, hyperparameter selection usually involves a grid search methodology, which can suffer from mesh size selection as well as an unknown optimal hyperparameter location leading to an uncertain region to be covered by the mesh.

There has also been significant work in the area of structure or network structure identification. The usage of other information based metrics is/has been performed with work focusing on the Transfer Entropy (a precursor of the Causation Entropy metric) [30, 31] and more recently the Directed Information [32, 33], which is a generalization of the Causation Entropy for non-Markovian processes. However, both these techniques have concerns as the Transfer Entropy has been demonstrated to incorrectly identify model structure when there is indirect coupling between features and Directed Information is difficult to estimate in actuality with almost no studies done on practical systems including considerations of noise or other issues that arise with practical applications [34]. This work seeks to demonstrate that the causation entropy metric is computationally feasible while allowing for accurate covariate selection for mechanical systems (which by nature can be considered Markovian for an appropriately selected sampling rate) to allow for improved black box and grey box modeling of actual systems by improving on MLE results by informing the optimization problem with an accurate model structure.

1.1.2 CEM Purpose

This work seeks to demonstrate the benefits of the CEM for covariate selection for a class of problems that can be defined as Markovian, nonlinear systems with a linear parameteri-

zation as shown Equation (1.1).

$$\begin{bmatrix} x_{t+1}^{(1)} \\ x_{t+1}^{(2)} \\ \vdots \\ x_{t+1}^{(n)} \end{bmatrix} = \begin{bmatrix} \theta_{11} & \cdots & \theta_{1,m} \\ \theta_{21} & \cdots & \theta_{2,m} \\ \vdots & \ddots & \vdots \\ \theta_{n1} & \cdots & \theta_{n,m} \end{bmatrix} \begin{bmatrix} f_1(x_t^{(1)}, ..., x_t^{(n)}, t) \\ f_2(x_t^{(1)}, ..., x_t^{(n)}, t) \\ \vdots \\ f_m(x_t^{(1)}, ..., x_t^{(n)}, t) \end{bmatrix} = \Theta * \mathbf{F}(\mathbf{X}_t, t) \tag{1.1}$$

The CEM is proposed as a method to identify the zero and nonzero parameters in the Θ matrix; any parameters identified as zero in the CEM imply that the corresponding function can be removed from the optimization problem of the corresponding state. This work will demonstrate that the CEM provides an improvement over existing techniques as computation of the CEM is deterministic and does not require comprehensive searching of any subsets as wrapper methods do. Additionally, the CEM is better able to identify the coupling between parameters to provide more accurate covariate selection than one at a time parameter methods available through common filter methods. Additionally, this work will demonstrate that the CEM identifies the system structure with greater accuracy than and with access to less training information, and no need for any sort of validation set, than LASSO and elastic net do. The ability to identify system structure in cases of limited training data is particularly important as a low data points to number of covariates ratio leads to an increased risk of overfitting. Thus, the CEM provides demonstrable benefits over existing covariate selection techniques for the class of problem considered.

1.2 Work Overview and Outline

This work explores the validity of using the Causation Entropy Matrix for the system identification, provides insights into necessary considerations for successful implementation of the technique, provides a comparison of results to current cutting edge algorithms and concludes with experimental results on a physical system. This work begins by providing necessary background on information theory and the definition of the Causation Entropy

Matrix (CEM) in the remainder of this chapter. Chapter 2 provides information about the algorithm used to both accurately estimate causation entropy as well as the underlying distribution. Chapter 3 demonstrates the applicability of the CEM to grey box system ID problems including an in depth explanation of CEM generation and computation. Chapter 3 also explores a connection between the the meaning of nonzero causation entropy values and a well known parameter metric known as the sensitivity.

Chapter 4 explores the applicability of the CEM to black box modeling tasks. The section also includes a comparison of the CEM technique to current state of the art techniques for sparse regression of LASSO and elastic net. Chapter 5 delves into necessary considerations for CEM usage and interpretation of results. First, the section explores the effects of noise on CEM performance and a corresponding discussion on the potential for use of filtering to combat noise effects. The chapter then has a discussion of some of the consequences of using KDE to estimate the underlying distributions. This includes a discussion of bandwidth selection, the curse of dimensionality and how to select the amount of data to include form available data to optimize CEM performance. Chapter 6 includes experimental results of application of the CEM technique to sensor data collected from a nonlinear system. Finally, the work concludes with Chapter 7, which provides final conclusions and some potential avenues for future work to explore.

1.3 Information Theory Background

The basic foundation of information theory lies in the notion of Shannon entropy, which is a measure of randomness in a signal or time series. Let $p(y,t)$ be the probability mass function of a stochastic process $Y(t) = [Y_1, ..., Y_n]^T$ at time t. The probability mass represents the probability that $Y = y$ for a given realization of Y. The entropy of the random process $p(y,t)$ at time t is defined by Shannon [35] according to,

$$H(Y) = -\sum p(y,t)\log(p(y,t)) \tag{1.2}$$

Processes with higher Shannon entropy appear more "random", while processes whose output is deterministic are defined to have zero entropy.

Let X and Y be two dependent, vector-valued random variables with joint mass $p_{XY}(\mathbf{x},\mathbf{y})$ and marginal mass $p_X(x)$ and $p_Y(y)$.

A joint entropy of X and Y is defined as in Eq. (1.3), which if X and Y are considered a multivariate distribution would result back in Eq. (1.2).

$$H(X,Y) = -\sum_{y}\sum_{x} p_{xy}(x,y) log(p_{xy}(x,y)) \tag{1.3}$$

The conditional distribution of Y given X is $p(y|x)$. The conditional entropy is defined as in Eq. (1.4). Conceptually, conditional entropy quantifies the amount of information needed to describe Y given knowledge of X.

$$H(Y|X) = -\sum\sum p_{xy}(x,y)\log(p(y|x)) \tag{1.4}$$

The entropy of a variable represents the the amount of information contained or uncertainty of a random variable. Thus, entropies are additive as shown below in Eq. (1.5), which is commonly referred to as the entropy chain rule [36].

$$H(X,Y) = H(X) + H(Y|X) \tag{1.5}$$

The mutual information defines the amount of information provided about X through observation of Y and is given by Eq. (1.6),

$$I(X;Y) = \sum_{x,y} p_{xy}(x,y) log\left(\frac{p_{xy}(x,y)}{p_x(x)p_y(y)}\right) \tag{1.6}$$

The definition for mutual information given in Eq. (1.6) is equivalent (with proof available in [36]) to the definition in Eq. (1.7), which intuitively matches the description of the

quantity mutual information represents.

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \tag{1.7}$$

Note that mutual information is thus symmetric with X and Y providing equal amounts of information about each other.

Two higher order information theoretic quantities particularly pertinent to the work are described here: transfer entropy and causation entropy. Let X and Y now represent continuous scalar random variables sampled at a certain rate, and let X_{t+1} and Y_{t+1} represent samples of X and Y at time $t+1$.

Now, let the past τ_x states of X be given by the vector,

$$X_t^{(\tau_x)} = (X_t, X_{t-1}, ..., X_{t-\tau_x+1}) \tag{1.8}$$

with a similar definition for $Y_t^{(\tau_x)}$. Then the transfer entropy from Y to X is given by Eq. (1.9) [37],

$$T_{Y\to X}^{\tau_x} = H(X_{t+1}|X_t^{(\tau_x)}) - H(X_{t+1}|X_t^{(\tau_x)}, Y_t^{(\tau_x)}) \tag{1.9}$$

Transfer entropy describes the extra information provided by Y_t in determination of X_{t+1}, in addition to that provided by X_t. The transfer entropy metric was originally proposed in [37]; however, Sun and Bollt [18, 38] showed that in the case of more than two variables that can have indirect coupling, the transfer entropy can identify incorrect relationships due to the ignoring of other states. Thus, indirect influences will be identified as direct influences by the transfer entropy metric.

Now consider a third stochastic process Z which interacts with X and Y. Causation entropy is a generalization of transfer entropy defined as [18],

$$C_{Z\to X|(X,Y)} = H(X_{t+1}|X_t^{(\tau_x)}, Y_t^{(\tau_x)}) - H(X_{t+1}|X_t^{(\tau_x)}, Y_t^{(\tau_x)}, Z_t^{(\tau_x)}) \tag{1.10}$$

In Eq. (1.10), $C_{Z\to X|(X,Y)}$ describes the causation entropy from Z to X conditioned on previous states of X and Y. This measures the amount of information provided to X from Z in addition to that provided by other means (i.e., from X itself and from Y). Note that transfer and causation entropies provide a single measure quantifying the amount of information transferred from one state to another, both in terms of magnitude and direction. For the purposes of this work, only $\tau_x = \tau_y = 1$ is considered, which thus assumes a Markovian process. Thus, the notation τ_x is omitted. Figure 1.2 gives a visual representation of the meaning of the causation entropy. The causation entropy is nonzero if and only if there is direct, causal information flow between two random variables. Note that transfer entropy would have returned all nonzero values as it cannot distinguish between direct and indirect influences and resultingly would have highlighted a relationship between Z_t and X_{t+1} even on the right side flow diagram of Figure 1.2.

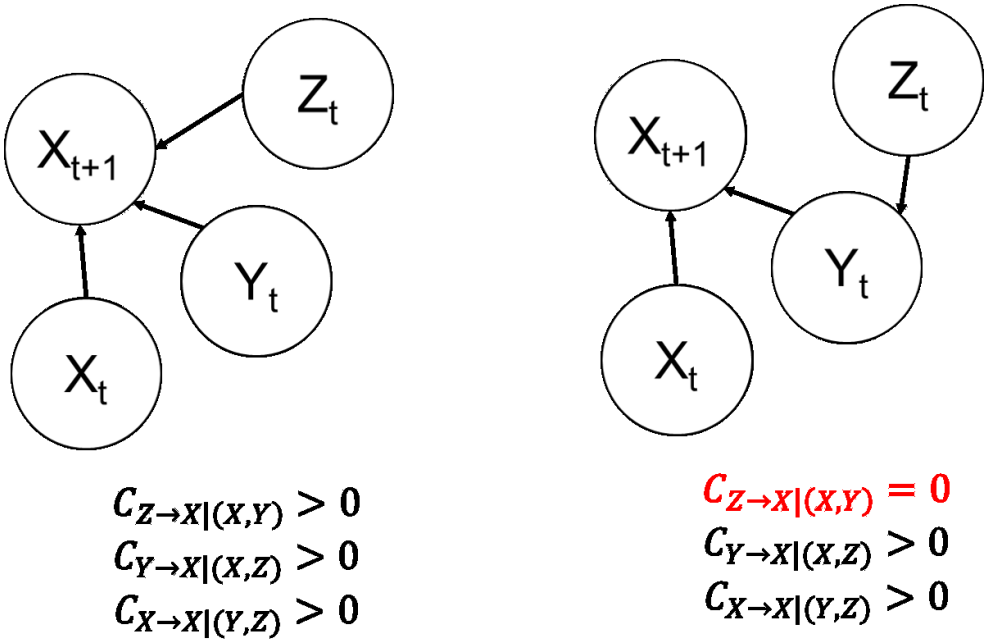

Figure 1.2: Visual representation of causation entropy

It is important to note that even though the above information theoretic quantities are computed as a function of probability masses of random variables, they can be suitably applied to quantities that are not random. In the context of this paper, the time histories

of the states are treated as random variables, and the underlying probability densities that generated them are estimated through the kernel density estimation process. By comparing certain conditional and joint probability densities of the state variables, the information flow between them is revealed. The relationship between the conditional probability densities, captured in Eq. (1.10), is used in the CEM to identify how the state components are influenced by each term in the dynamic equation. Thus, by treating the state time histories as realizations of random variables, the underlying distributions can be estimated and processed using information theoretic tools to reveal the structure of the dynamical system.

1.3.1 Causation Entropy Matrix (CEM) Definition

Previous work by Kim *et al* [17] showed that causation entropy can be naturally applied to parameter estimation problems for a certain class of dynamical systems by considering the measured time series data as a sequence of realizations of random variables. This was a natural extension of the network structure identification methods proposed by Sun *et al* in [39, 40]. To understand how this works, consider a discrete time, nonlinear, time varying system expressed in the form of Eq. (1.11),

$$\begin{bmatrix} x_{t+1}^{(1)} \\ x_{t+1}^{(2)} \\ \vdots \\ x_{t+1}^{(n)} \end{bmatrix} = \begin{bmatrix} \theta_{11} & \cdots & \theta_{1,m} \\ \theta_{21} & \cdots & \theta_{2,m} \\ \vdots & \ddots & \vdots \\ \theta_{n1} & \cdots & \theta_{n,m} \end{bmatrix} \begin{bmatrix} f_1(x_t^{(1)}, ..., x_t^{(n)}, t) \\ f_2(x_t^{(1)}, ..., x_t^{(n)}, t) \\ \vdots \\ f_m(x_t^{(1)}, ..., x_t^{(n)}, t) \end{bmatrix} = \Theta * \mathbf{F}(\mathbf{X}_t, t) \tag{1.11}$$

where $x^{(j)}$ denotes the j^{th} element of the state vector $\mathbf{X}$ and $x_t^{(j)}$ denotes the value of $x^{(j)}$ at timestep t. Note that Θ is an $n \times m$ matrix of parameter values (which are assumed to be constant over the time scale of the data used in parameter estimation) and $\mathbf{F}$ is an $m \times 1$ vector. By computing the causation entropy of f_i on $x_t^{(j)}$, conditioned on all $\mathbf{F}$ except f_i (denoted herein as $\mathbf{F} \backslash f_i$), insight can be gained into the importance of parameter θ_{ij} in

driving the system dynamics. For instance, if $\theta_{ij} = 0$, then the causation entropy of f_i on $x^{(j)}$ conditioned on the other elements of $\mathbf{F}$ would be zero, since f_i does not contribute any information to the random process realized by $x^{(j)}$. Likewise, if θ_{ij} is relatively large, then a significant amount of information transfer occurs between f_i and $x^{(j)}$ and the corresponding causation entropy would be expected to be large. In this way, causation entropy can be used to identify the underlying parametric structure of a system from data realizations.

To formalize this, define a matrix, referred to herein as the *Causation Entropy Matrix* (CEM), as

$$CEM = \begin{bmatrix} C_{f_1 \to x^{(1)}|[\mathbf{F}\setminus\mathbf{F}^{(1)}]} & C_{f_2 \to x^{(1)}|[\mathbf{F}\setminus\mathbf{F}^{(2)}]} & \cdots & C_{f_m \to x^{(1)}|[\mathbf{F}\setminus\mathbf{F}^{(m)}]} \\ C_{f_1 \to x^{(2)}|[\mathbf{F}\setminus\mathbf{F}^{(1)}]} & C_{f_2 \to x^{(2)}|[\mathbf{F}\setminus\mathbf{F}^{(2)}]} & \cdots & C_{f_m \to x^{(2)}|[\mathbf{F}\setminus\mathbf{F}^{(m)}]} \\ \vdots & \vdots & \ddots & \vdots \\ C_{f_1 \to x^{(n)}|[\mathbf{F}\setminus\mathbf{F}^{(1)}]} & C_{f_2 \to x^{(n)}|[\mathbf{F}\setminus\mathbf{F}^{(2)}]} & \cdots & C_{f_m \to x^{(n)}|[\mathbf{F}\setminus\mathbf{F}^{(m)}]} \end{bmatrix} \tag{1.12}$$

This Causation Entropy Matrix can be applied to any nonlinear system which can be expressed in the form of Eq. (1.11). The matrix is structured so that each entry gives the amount of information (in bits or nats depending on the log base used) that a given element of $\mathbf{F}$ provides to a state update in addition to that provided by other elements of $\mathbf{F}$. In the case where an entry in the CEM is computed to be zero, the corresponding function provides no additional information to a particular state beyond what is already contained within the other functions. Therefore, the corresponding parameter in Θ should be zero. Note that for a linear system, $\mathbf{F} \equiv \mathbf{X}$, which means that the CEM will collapse to an $n \times n$ square matrix. A detailed discussion of linear systems, including a closed form solution for CEM in the linear case, is provided in [17] and [38].

The expression for the CEM in Equation (1.12) is far more general than that proposed previously in [17]. In [17], the formulation for CEM required that entries of $\mathbf{F}$ be of the form $f_i(x_j)$, which means that each element of $\mathbf{F}$ is a function of only one state and not explicitly dependent on time. This requirement placed a significant limitation on the type of

systems to which the Causation Entropy Matrix could be applied. In this work it is shown that such a restriction is unnecessary and the matrix can be generalized to systems of the form shown in (1.11) with no penalty in performance or accuracy.

CHAPTER 2

CAUSATION ENTROPY MATRIX COMPUTATION AND ESTIMATION

This chapter seeks to explore the ideas and algorithms that allow for the actual implementation and estimation of the entropy values for use in various system identification tasks. This includes discretizing continuous models to allow for the model form to match that required for the CEM, estimating the underlying probabilities necessary for entropy calculation, the computational method used to accurately compute the entropy from sampled and estimated probabilities, and finally a discussion of probability mass versus density as it applies to this work.

2.1 Model Discretization

Most mechanical systems are governed by continuous time equations of motions. However, all definitions of entropy metrics, including the causation entropy, were made for discrete time systems. Mechanical systems are typically governed by continuous time ordinary differential equations; In order to be able to use the CEM on a mechanical system, a discretization method must be used. This work uses a forward finite-difference approximation to the derivative as shown below in Eq. (2.1).

$$\dot{x}^{(i)} = \frac{x_{t+1}^{(i)} - x_t^{(i)}}{T} \tag{2.1}$$

T in Eq. (2.1) is the time step used in the finite difference approximation. In order to be used with the CEM, T must be chosen to be constant and to be significantly smaller than the rate of change of dynamics of the system. Equation (2.2) represents a reordering of Eq. (2.1) that allows for the discretization of the continuous time system into a discrete one that

can be used in conjunction with the CEM.

$$x_{t+1}^{(i)} = T\dot{x}^{(i)} + x_t^{(i)} \tag{2.2}$$

A more in depth look at the requirements and impact of the selection of the time step T is included in Chapter 5.1.

2.2 Density and Entropy Estimation

2.2.1 Kernel Density Estimation

Throughout this paper, estimates of causation entropy are presented for various example time series data. Numerous methods have been proposed for numerically estimating joint entropy, including kernel density estimation (KDE) [39, 40] and k-nearest neighbor algorithms [41]. All results in this work estimate joint entropy using the KDE approach outlined in [17], which is based on the KDE estimator proposed in [42].

Kernel density estimation operates by attempting to approximate the underlying probability density function (PDF) $f(x)$ based off of observed data points [43]. It accomplishes this task by placing a kernel (in this work Gaussian kernels are used) at each observed data point, and then summing the contributions of each kernel to create a composite PDF. Thus, areas with multiple nearby data points will have higher support than areas with sparse or little information. The equations that govern the KDE scheme used are given in (2.3-2.5). Consider a matrix of observed data Y with n observations and a random vector X to determine the probability density of from Y. S is the covariance matrix of Y.

$$f(X) = \frac{1}{n}\sum_{i=1}^{n} K(u) \tag{2.3}$$

In Eq. (2.3), $K(u)$ represents the kernel to be used to generate the composite PDF evaluated at u. u, given in Equation (2.4), provides a measure of the distance between X

and the data Y.

$$u = \frac{(X - Y_i)^T S^{-1} (X - Y_i)}{h^2} \tag{2.4}$$

Eq. (2.5) provides the equation for the Gaussian kernel.

$$K(x) = \frac{1}{(2 * \pi)^{\frac{d}{2}} h^d det(S)^{\frac{1}{2}}} exp(-\frac{x}{2}) \tag{2.5}$$

In Equations (2.4) and (2.5), h is the estimator bandwidth. The bandwidth is a smoothing parameter that needs to be selected in order for the KDE to be completed [44]. The bandwidth relates to the space around data points to be included and with what density. Intuitively, a small bandwidth will lead (in 2D) to a tall and skinny Gaussian, which corresponds to a very high probability that the data is very close to the observed data point. A larger bandwidth corresponds (in 2D) to a shorter, wider Gaussian, which will lead to a more evenly distributed probability distribution over a wider area around the data point.

$$h = (\frac{4}{d+2})^{\frac{1}{(d+4)}} * (N)^{\frac{-1}{(d+4)}} \tag{2.6}$$

In Eq. (2.6), d is the dimension of the data and N is the number of data points included. Equation (2.6) is an automatic bandwidth selection rule that removes the need for the user to tune any parameters in order to perform KDE (and thus estimate causation entropy).

It is important to note that the causation entropy estimation approach used here can be computationally burdensome for large datasets and for high-dimensional data. The computation of the covariance matrix, its inverse and determinant are all computationally expensive. It is then required that all data points are summed over, which can lead to a large computational load in order to simply estimate the PDF, which is the main reason why the CEM cannot be used in real-time applications. In addition to the large computational burden of high dimensional datasets, kernel density estimation, like most machine learning techniques, suffer in higher dimension from the curse of dimensionality. The impact

of the curse of dimensionality on KDE is explored in detail in (5.2.2). The large computational load and issues with the curse of dimensionality warranted an exploration of a different method for density estimation and entropy estimation. Thus, an algorithm based on the common technique of K-nearest-neighbors was also attempted and results discussed in 2.2.4.

2.2.2 Shannon Entropy Estimation

The techniques in section 2.2.1 provide a means for estimating the underlying PDF needed for entropy estimation. However, the basic equations for entropies in (1.2, 1.3, and 1.4) will not be effective if directly using the estimated probability densities. If one tried to sum over the entirety of the estimated PDF, there would be areas that are sparse or have little support; however, this may not be because the actual, true PDF is sparse but simply because data in the region has not been encountered. Thus, results might be overly skewed if the entirety of the estimated PDF is used. Therefore, a resubstitution, plugin estimator is used to compute entropy as shown in Eq. (2.7) [45, 46].

$$\hat{H}(x) = -\frac{1}{n}\sum_{i=1}^{n} log(\hat{p}(x)) \tag{2.7}$$

$\hat{H}(x)$ is the estimate of the entropy and $\hat{p}(x)$ is the estimate of the probability mass function of x. A resubstitution estimator uses all observed data from X i.e.$[X_1, ..., X_n]$ to estimate the probability density. The plugin estimator only sums contributions to the entropy value where a data point has been observed. Thus, the PDF will only be sampled in order to calculate the causation entropy at observed data points, which by definition will be non sparse sections of the PDF.

2.2.3 From Probability Density to Probability Mass

Thus far, there has been little distinction made between Probability Density Functions (PDF) and Probability Mass Function (PMF). A probability density function is a continuous function, where the integral between any two points yields the probability of an event occurring within said range as the the probability of any exact event happening from a continuous distribution is zero. A PMF represents the probability of an event happening from a discrete distribution. The probability of any possible, specific event occurring from a discrete distribution is certainly nonzero [47]. The above discussion of KDE in Section 2.2.1 will yield a continuous PDF. However, the entropy definitions in Equations (1.2-1.4), which are the basis for all future derived entropy metrics assume a discrete distribution and a corresponding PMF. It may appear as though the PDF is merely being used in place of a PMF. In general, this practice is incorrect and will lead to improper results as PDFs and PMFs have very different characteristics with the most obvious, but certainly not the only, being that PMF values are by definition bounded between $[0, 1]$, whereas a PDF has no such restriction as the only requirement is that the PDF is greater than or equal to 0 and integrate to 1.

This section will take some time to detail how the transition between the two is made to demonstrate the validity of the method described. If one considers KDE, the PDF is entirely defined by the observed data and the bandwidth used. However, if one wishes to make a discrete representation of the PDF, one must discretize the PDF by sampling it at discrete points and creating bins or a mesh around each point and assuming a uniform value for the PDF within each bin. For this work, a simple rectangle rule is used for the shape of each bin [48]. Assume that constant sized bins are used with uniform length along each axis ΔM. The function can then be integrated with the value of the "volume" of the bin used. For this section, volume refers to the space underneath a point of the PDF. For a 2D PDF, the mesh will be an actual volume as in Figure 2.1, but in higher, N- dimensional systems the "volume" will actually be the $(\Delta M)^N \times (PDFvalue)$. In order to demonstrate

the validity of practice, an example is given in Figure 2.1.

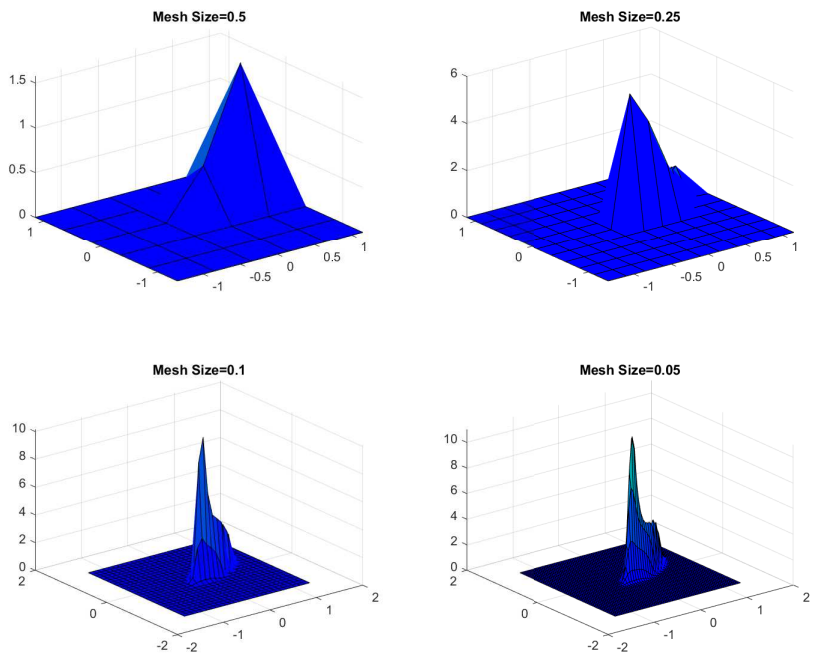

Figure 2.1: Visual representation of mesh size on PDF discretization

Data was generated by a simple second order linear system. Kernel density estimation was then performed using the bandwidth rule from Eq. (2.6). In each of the four subplots, the data is the same, but the PDF is generated with a different size mesh. One can see that as the mesh becomes finer, the PDF appears smoother. The numerical integral over the PDF is then computed for each with the results of (in order of decreasing mesh size) 0.5602, 1.0423, 1.0000 and 1.0000 respectively. The integral over a PDF should be equal to one as if you compute the probability of an event occurring amongst all possible outcomes, the probability should be 1. The functions with values corresponding to mesh sizes 0.1 and 0.05 satisfy the requirements for a PDF to integrate to one. Similarly, one could consider the discretized PDF to be a PMF where the discrete possibilities are the points of each mesh and the probability to be the mesh "volume". In this case, the integral has been reparameterized as a pure summation that satisfies the conditions for a PMF. Thus, a method for converting

a PDF to a PMF has been presented; however, it introduces a new auxiliary problem of how to select an appropriate mesh size to ensure an accurate PDF to PMF conversion. Fortunately, the characteristics of the nature of the problem considered actually remove this concern.

Consider the nature of the causation entropy estimation, where the causation entropy is defined as in Eq. (2.8). Consider the case where a single mesh size ΔM is used for all calculations.

$$CE_{Z\to X|Y} = H(X|Y) - H(X|Y,Z) \tag{2.8}$$

Bayes law, given by Equation (2.9) is used to compute the conditional probabilities needed for causation entropy estimation [47].

$$p(A|B) = \frac{p(A,B)}{p(B)} \tag{2.9}$$

Thus, using the plugin estimator, the necessary calculations are shown below in Equation (2.10)-(2.11).

$$\begin{aligned}
\hat{CE}_{Z\to X|Y} &= H(X|Y) - H(X|Y,Z) \\
&= \frac{1}{n}\left[-\sum_{i=1}^{n}\left(\log\left(\frac{p_{xy}(x,y)}{p_y(y)}\right)\right) - \left(-\sum_{i=1}^{n}\left(\log\left(\frac{p_{xyz}(x,y,z)}{p_{yz}(y,z)}\right)\right)\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n}\left(\log\left(\frac{p_y(y)}{p_{xy}(x,y)}\right)\right) - \sum_{i=1}^{n}\left(\log\left(\frac{p_{yz}(y,z)}{p_{xyz}(x,y,z)}\right)\right)\right]
\end{aligned} \tag{2.10}$$

Now, consider the fact that when computing the CEM, X (the state) and Z (the potential function being considered) has only one dimension, therefore p_{xy} is in a one dimensional higher space than p_y and similarly so for p_{xyz} and p_{yz}. Now, consider f to be the probability density function, and without loss of generality assume that Y is also one dimensional.

Note that the proof proceeds identically if Y has a higher dimension.

$$\begin{aligned}
\hat{CE}_{Z\to X|Y} &= \frac{1}{n}\left[\sum_{i=1}^{n}\left(\log\left(\frac{f_y(y)\Delta M}{f_{xy}(x,y)(\Delta M)^2}\right)\right) - \sum_{i=1}^{n}\left(\log\left(\frac{f_{yz}(y,z)(\Delta M)^2}{f_{xyz}(x,y,z)(\Delta M)^3}\right)\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n}\left(\log\left(\frac{f_y(y)}{f_{xy}(x,y)\Delta M}\right)\right) - \sum_{i=1}^{n}\left(\log\left(\frac{f_{yz}(y,z)}{f_{xyz}(x,y,z)\Delta M}\right)\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n}\left(\log\left(\frac{f_y(y)}{f_{xy}(x,y)}\times\frac{1}{\Delta M}\right)\right) - \sum_{i=1}^{n}\left(\log\left(\frac{f_{yz}(y,z)}{f_{xyz}(x,y,z)}\times\frac{1}{\Delta M}\right)\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n}\left(\log\left(\frac{f_y(y)}{f_{xy}(x,y)}\right)+\log\left(\frac{1}{\Delta M}\right)\right)\right] \\
&\qquad - \frac{1}{n}\left[\sum_{i=1}^{n}\left(\log\left(\frac{f_{yz}(y,z)}{f_{xyz}(x,y,z)}\right)+\log\left(\frac{1}{\Delta M}\right)\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n}\left(\log\left(\frac{f_y(y)}{f_{xy}(x,y)}\right)-\log\left(\frac{f_{yz}(y,z)}{f_{xyz}(x,y,z)}\right)\right)\right] \\
&\qquad + \frac{1}{n}\left[\sum_{i=1}^{n}\left(\log\left(\frac{1}{\Delta M}\right)-\log\left(\frac{1}{\Delta M}\right)\right)\right] \\
&= \frac{1}{n}\left[\sum_{i=1}^{n}\left(\log\left(\frac{f_y(y)}{f_{xy}(x,y)}\right)\right)\right] - \frac{1}{n}\left[\sum_{i=1}^{n}\left(\log\left(\frac{f_{yz}(y,z)}{f_{xyz}(x,y,z)}\right)\right)\right]
\end{aligned} \tag{2.11}$$

Notice that the final results of Equations (2.10) and (2.11) are identical with the exception of the PMFs from Equation (2.10) replaced with PDFs in Equation (2.11). Thus, for this one application, it is acceptable to use the PDF values returned directly from kernel density estimation directly in the entropy estimation from Equation (2.7), as there is an underlying assumption that a small enough mesh theoretically exists that allows for the discretization of the PDF into a viable PMF. Inclusion of an appropriate, constant mesh size has no impact on the overall result of causation entropy estimation.

2.2.4 K Nearest Neighbors

In addition to the KDE work done here, an attempt was made to explore the KNN estimator proposed in [41]. The goal of creating a working second estimator was to be able to have a potentially faster or more robust estimator based on a separate principle. KNN estimators are known to incredibly fast when the data set is small and provide a method to estimate underlying probability distributions based upon the number and distance to the nearest neighboring data points [49, 50].

In [41], the following KNN estimator for entropy $\hat{H}(x)$ in Equation (2.12)

$$\hat{H}(x) = -\psi(k) + \psi(n) + \log(c_d) + \frac{d}{n}\sum_{i=1}^{n} \log \epsilon(i) \tag{2.12}$$

In Equation (2.12), ψ is the digamma function, k is the number of nearest neighbors to be considered (a number that must be selected by the user), n is the number of data points used, d is the dimension of the data, c_d is the volume of the d-dimensional unit ball and $\epsilon(i)$ is twice the distance from x_i to its k-th nearest neighbor. Derivation and further details on the algorithm can be found in [41].

The above algorithm was implemented and tested. The algorithm was validated by generating data from a normal distribution with a given mean (μ) and covariance (Σ). This distribution was chosen as the actual value of the the entropy of a normal distribution is known and given by Equation (2.13) [51].

$$H_{actual}(X) = \frac{1}{2}(1 + \log(2\pi|\Sigma|)) \tag{2.13}$$

Two dimensional data was generated with $100,000$ data points from a random distribution with the following parameters:

$$\Sigma = \begin{bmatrix} 14 & 7 \\ 7 & 12 \end{bmatrix}, \quad \mu = \begin{bmatrix} 1 \\ 4 \end{bmatrix}$$

The exact entropy is 5.227 and the estimated entropy with $k = 4$ was computed as 5.225. Thus, for this low dimensional, high number of data points case the KNN estimator was very accurate in estimating the entropy.

However, in order to be useful in this work, it is necessary to determine whether the estimator is successful in estimating the causation entropy. In order to test this, the causation entropy needs to be computed.

Using the chain rule from Equation (1.5), the Causation Entropy can be rewritten as four joint entropies instead of two conditional entropies as shown in Equation (2.14).

$$\begin{aligned} CE_{Z\to X|S} &= H(X|S) - H(X|S,Z) \\ &= H(X,S) + H(Z,S) - H(X,Z,S) - H(S) \end{aligned} \tag{2.14}$$

Each joint distribution can be considered a single multivariate distribution and thus Equation (2.12) can be used to compute the CEM.

To test this simply, a pendulum system with damping and harmonic excitation was considered with equations of motion given in Equations (2.15-2.16). More details on the process of CEM computation for grey box systems is included in Chapter 3.1; the results are merely shown here to demonstrate the potential of the KNN estimator. m is the mass of the pendulum, l the length of the arm, and c is the rotational damping coefficient.

$$\dot{x}_1 = x_2 \tag{2.15}$$

$$\dot{x}_2 = -\frac{g}{l}sin(x_1) - \frac{c}{ml^2}x_2 + 3sin(3t) \tag{2.16}$$

Thus, the CEM can be computed and should have the structure given below by $Model_{actual}$

$$Model_{actual} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}$$

The CEM was then computed using both kernel density estimation and the KNN algorithm with results shown below.

$$CEM_{KNN} = \begin{bmatrix} 0.0076 & 0 & 0.0083 & 0 \\ 0 & 0.1574 & 0 & 0 \end{bmatrix} \tag{2.17}$$

$$CEM_{KDE} = \begin{bmatrix} 0 & 13.9628 & 12.1512 & 0 \\ 7.4651 & 17.5842 & 0 & 4.4328 \end{bmatrix} \tag{2.18}$$

Clearly, the KDE based CEM was able to identify the structure whereas the KNN based CEM was not. There are multiple potential causes for the pitfalls of the KNN estimator. The first is that in higher dimensional situations, it is very difficult to populate the space to have a nearby neighbor and the distances between points become difficult to calculate and can be less meaningful predictors [52]. Additionally, the potentially different scaling of the data can become problematic in estimating the underlying PDF [53] as the distances generate distorted results as units and the range of values between variables can be rather different [50]. Additionally, the KNN estimator introduces the problem of having to select the the number of neighbors to consider as well as potential concerns about the scaling of the data, which may require additional pre processing to prepare for use. Thus, the KDE method was used for this work, but further exploration of the KNN method and its application to causation entropy is certainly an interesting area to explore, though outside the scope of this work.

2.3 Permutation Test

One of the main goals of the CEM is to identify parameters that should be zero, which allows for a reduction in the size of the parameter set that must be estimated. This reduces the dimensionality of the resulting numerical optimization problem, with convergence benefits shown in [17]. However, entropy estimation biases due to finite precision, finite data length, and noise in measured data, means that a causation entropy estimate will never be identically zero even if it theoretically should be. Therefore, a statistical test is needed to determine if a value in CEM should actually be zero. One common statistical test used for this purpose is a permutation test [54], which compares the estimated value for the causation entropy with causation entropy values computed from randomly permuted time series data. If the computed CEM value is not greater than p percent of randomly computed test causation entropies, then the entry is statistically nonzero. In this work, p is selected as 99%. Use of the permutation test to identify thresholds for a zero causation entropy was proposed in [55] and used effectively in [17]. In this work, on the order of $n = 100$ permutations of the data was used in order to test for significance. Additionally, if the estimator returned a negative value, this entry was assumed to be zero as a negative causation entropy is not defined as information removal by a potential function is not possible.

It is worth noting, that the permutation test is the most time consuming portion of the computation of the CEM as for each entry in the CEM n additional causation entropies must be computed. Unfortunately, the results are not reusable in any way, thus time scales linearly with the number of permutations desired. As will be discussed later in Sections 3.2 and 4.1.3, there is meaning from the relative magnitudes of the nonzero CEM entries, and thus, if computation speed is an issue, the number of permutations can be reduced or the test removed entirely with more choice left to the user. Through the experiments run in this work and observing the pre and post permutation test CEM values, entries that are removed by the permutation test often had magnitudes at least one order of magnitude less

than those that had meaning and should be nonzero or were negative (which is undefined and suggests 0 but with an estimation error included); thus these unnecessary parameters could often be removed by inspection.

CHAPTER 3

APPLICATION OF CAUSATION ENTROPY MATRIX TO PHYSICAL SYSTEMS

This chapter seeks to demonstrate the applicability of the CEM for grey-box model system ID tasks. Section 3.1.1 provides an in depth description of how to perform model discretization and CEM formulation as well as necessary calculations for application to grey box problems. Sections 3.1.2 and 3.1.3 provide more complex examples of CEM application as well as a demonstration of application to time varying systems. The chapter concludes with Section 3.2, which discusses the meaning that can be drawn from the nonzero entries in the CEM.

3.1 Grey-Box System Identification

3.1.1 Pendulum Example

The first example considers a pendulum created by hanging a point mass of mass m a distance L from a pivot with viscous damping c at the pivot. Letting $x^{(1)}$ denote the angular displacement and $x^{(2)}$ denote the angular velocity, the state space representation of the equations of motion, given by Equations (3.1) and (3.2), can be generated by a torque summation about the pivot.

$$\dot{x}^{(1)} = x^{(2)} \tag{3.1}$$

$$\dot{x}^{(2)} = -\frac{g}{L}\sin(x^{(1)}) - \frac{c}{mL^2}x^{(2)} \tag{3.2}$$

Using Equation (2.2), Equations (3.1) and (3.2) can be transformed into a discrete time system given by Equations (3.3) and (3.4).

$$x_{t+1}^{(1)} = Tx_t^{(2)} + x_t^{(1)} \tag{3.3}$$

$$x_{t+1}^{(2)} = T(-\frac{g}{L}\sin(x_t^{(1)}) - \frac{c}{mL^2}x_t^{(2)}) + x_t^{(2)} \tag{3.4}$$

Equations (3.3) and (3.4) can be put into the form of (1.11) as follows:

$$\begin{bmatrix} x_{t+1}^{(1)} \\ x_{t+1}^{(2)} \end{bmatrix} = \begin{bmatrix} 0 & T & 1 \\ -\frac{Tg}{L} & (1 - \frac{Tc}{mL^2}) & 0 \end{bmatrix} \begin{bmatrix} \sin(x_t^{(1)}) \\ x_t^{(2)} \\ x_t^{(1)} \end{bmatrix} \tag{3.5}$$

CEM Computation Example

The causation entropy matrix can be applied directly to the system as represented in Equation (3.5). With $m = 1$ kg, $L = 1$ m, $c = 1.7$ N-m-s/rad, a time series realization of system states was produced from initial conditions $x_0^{(1)} = 0.2$ rad, $x_0^{(2)} = 0$ with a time step of $T = 0.01$ sec. A total of 400 datapoints (time steps) were generated by the discrete dynamic equations and recorded, and the CEM was computed using the plug-in estimator described in the previous section [56]. Simulated state data is available and can be separated into two sets: one from $t_{early} = [t_0, t_{f-1}]$ and one from $t_{late} = [t_1, t_f]$. Thus, using the state values from the set t_{early}, the values of the potential state functions at time step t, $\mathbf{F}(t_{early})$, can be computed. From this, the necessary input-output pairs can be generated to allow for the necessary KDE for CEM estimation.

CEM Computation Results

The computed value for the causation entropy matrix is

$$CEM = \begin{bmatrix} 0 & 7.00 & 4.20 \\ 0.948 & 10.60 & 0 \end{bmatrix} \tag{3.6}$$

This structure exactly matches the expected structure of Equation (3.5), with zeros in the top left and bottom right positions and nonzero values in all other positions. Thus, based only a time history of the recorded states, it is possible to estimate the structure of the parameter matrix prior to identification of the actual parameter values themselves.

The simple pendulum is an example of a mechanical system that requires conversion from continuous to discrete time; however, its dynamics are relatively simple and it does not violate the rather strict requirements for the system structure given previously in [17]. Subsequent examples of a pendulum on a cart and angle of attack dynamics of a projectile exhibit stronger nonlinearities and require use of the generalized form of CEM shown in Equation (1.12).

3.1.2 Pendulum on a Cart Example

Consider the classical pendulum on a cart system, in which a simple pendulum is mounted to a pivot on a moving cart. The continuous-time nonlinear equations of motion for this

system can be represented in the form of Equations (3.7) to (3.11) [57]:

$$\dot{x}^{(1)} = x^{(2)} \tag{3.7}$$

$$\dot{x}^{(2)} = \frac{1}{D_1}[-m^2L^2g\cos(x^{(3)})\sin(x^{(3)}) + \ldots \\ (u + mlx^{(4)^2}\sin(x^{(3)}) - dx^{(2)}(I + mL^2)] \tag{3.8}$$

$$\dot{x}^{(3)} = x^{(4)} \tag{3.9}$$

$$\dot{x}^{(4)} = \frac{1}{D_1}[(M+m)(mgL\sin(x^{(3)})) - \ldots \\ (u + mLx^{(4)^2}\sin(x^{(3)} - dx^{(2)}mL\cos(x^{(3)})] \tag{3.10}$$

$$D_1 = (1 + mL^2)(M+m) - m^2l^2\cos^2(x^{(3)}) \tag{3.11}$$

where $x^{(1)}$ and $x^{(2)}$ are the position and velocity of the cart respectively, and $x^{(3)}$ and $x^{(4)}$ are the angular position and angular velocity of the pendulum. In Equations (3.7) to (3.11), M is the mass of the cart, m is the mass of the pendulum, L is the pendulum length, g is the gravitational acceleration, d is a coefficient of viscous damping at the pivot point, and u is an input force applied to the cart. It is straightforward to apply Equation (2.2) to transform this system into a discrete time representation. When expressed in the form of Equation (1.11), the resulting discrete system will yield a Θ matrix of dimension 4×12 and $\mathbf{F}$ of dimension 12×1.

A simulated trajectory of this system was generated starting from initial conditions $x^{(1)} = 0$, $x^{(2)} = 0$, $x^{(3)} = \pi/6$ rad, $x^{(4)} = 0$ with a timestep of $T = 0.01$ sec and a constant input of $u = 50$ N. The CEM was computed from the resulting discrete system trajectory. Fig. 3.1 shows a shaded plot of the absolute value of the entries in the actual system parameter matrix Θ (left) and the corresponding causation entropy matrix (right). White corresponds to a zero entry in the matrix, while darker colors denote larger magnitudes (note that each plot has a different scale). The two figures identify nearly identical nonzero entries, and the entries corresponding to high-magnitude parameters (darkest in

color) are identically located in both the real system and the causation entropy matrix. The CEM has a 97.9% (47/48) covariate selection accuracy in correctly identifying whether an entry in the actual matrix was zero or nonzero. In the scope of this work, the covariate selection accuracy is defined as the accuracy correctly identifying if a feature should be included in a model. Thus, if an entry in the CEM is either correctly equal to zero or correctly nonzero, the entry is considered a success. The covariate selection accuracy is the number of successful entries in the CEM divided by the total number of entries in the CEM.

In the one instance where a value in CEM is nonzero when the parameter is in fact zero in Θ, the reported causation entropy is an order of magnitude smaller than all other values in the same row of CEM, which implies that the corresponding function provided significantly less (actually zero) information than all of the functions with nonzero parameters to the state update equation. While it is difficult to see given this small magnitude, the erroneous non-zero value in CEM is located at position (4, 1). Note that this example is clearly not additively separable in its states, meaning that it does not satisfy the requirements for CEM previously presented in [17] and requires the more generalized form provided in (1.12).

It is also interesting to consider performance of the CEM estimator in the presence of a time-varying forcing function. Consider again the pendulum on a cart, now including a sinusoidal input such that $u = \sin(\omega t)$ in Equation (3.8) and Equation (3.10). The time varying terms are incorporated into $\mathbf{F}$ in order to maintain Θ as a matrix of constants.

$$\mathbf{F}(X_t, t) = [x_2 \quad x_1 \quad \tfrac{cos(x_3)sin(x_3)}{D_1} \quad \tfrac{1}{D_1} \quad \tfrac{x_4^2 sin(x_3)}{D_1} \quad \tfrac{x_2}{D_1} \quad x_4 \quad x_3 \quad \tfrac{sin(x_3)}{D_1} \quad \tfrac{cos(x_3)}{D_1} \quad \tfrac{sin(x_3)cos(x_3)x_4^2}{D_1} \quad \tfrac{cos(x_3)x_2}{D_1}] \tag{3.12}$$

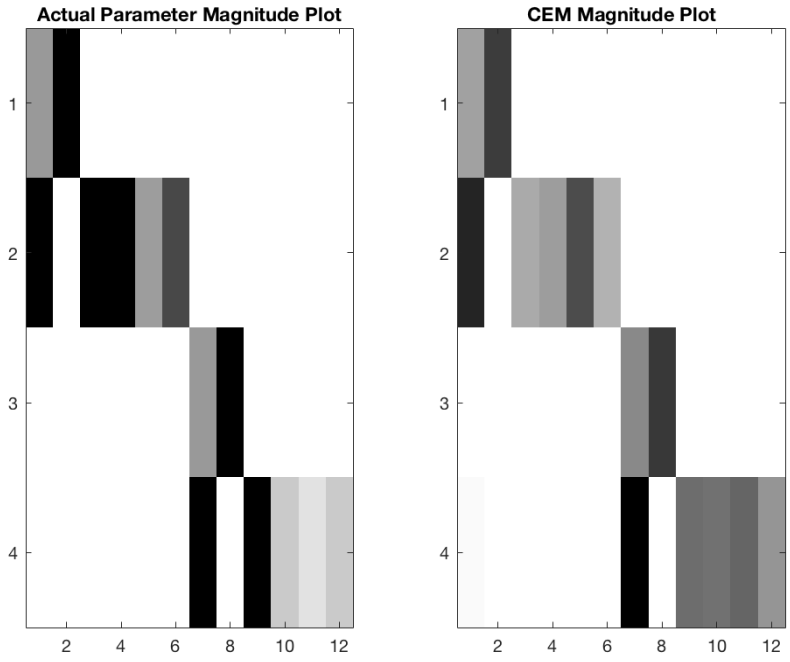

Figure 3.1: Magnitude plots of actual system matrix and estimated CEM

$$\mathbf{F}(X_t,t) = [x_2 \quad x_1 \quad \frac{cos(x_3)sin(x_3)}{D_1} \quad \frac{sin(\omega t)}{D_1} \quad \frac{x_4^2 sin(x_3)}{D_1} \quad \frac{x_2}{D_1} \quad x_4$$
$$x_3 \quad \frac{sin(x_3)}{D_1} \quad \frac{sin(\omega t)cos(x_3)}{D_1} \quad \frac{sin(x_3)cos(x_3)x_4^2}{D_1} \quad \frac{cos(x_3)x_2}{D_1}] \quad (3.13)$$

Figure 3.2 shows the time history of the system with the above physical parameters and initial conditions using a sinusoidal input with $\omega = 5$ rad/s. The time history shows that the discretized system (dashed line) maintains reasonable accuracy with respect to the continuous dynamics (solid line). The lines would converge even more if the timestep was decreased. Figure 3.3 looks nearly identical to Fig. 3.1, which is expected because the matrix Θ is unchanged by the introduction of a time varying forcing term. This can be visualized by the constant input and sinusoidal input state functions vectors $\mathbf{F}(X_t,t)$ given respectively in Equations (3.12) and (3.13). Notice that terms 4 and 10 in Equation (3.13) contain the forcing terms of the input not seen in Equation (3.12). Thus, the remaining parameters for the Θ matrix remain unchanged as all changes occur in $\mathbf{F}(X_t,t)$. Thus, the

zero and nonzero entry locations are identical and the relative magnitudes of the parameters are the same. For the simulation of the pendulum on a cart with sinusoidal input, the CEM had a 100% (48/48) accuracy in correctly identifying whether or not an entry in the actual matrix was zero or nonzero. Notice that in the case with time varying input, the CEM had improved accuracy in estimating the model structure. The inclusion of the excitation leads to a richer time series with all modes excited, which allows for better parameter identification. The effect of a rich excitation is well known on improving/effecting paramter excitation [58].

Unlike all previous examples here and in [17] which considered autonomous systems only, this example demonstrates that the CEM can be successfully applied to systems which are subject to time-varying forcing.

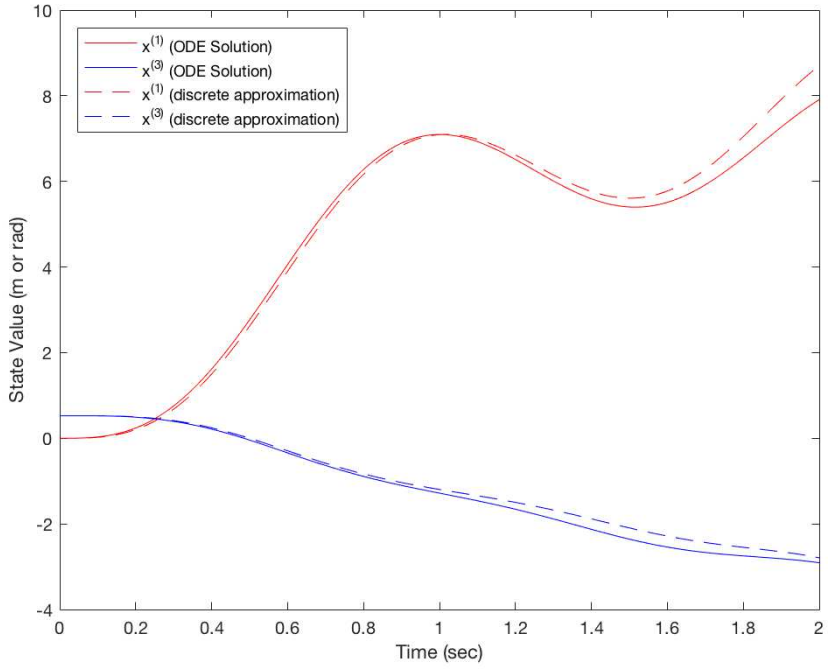

Figure 3.2: Time simulation of pendulum on cart with harmonic cart excitation

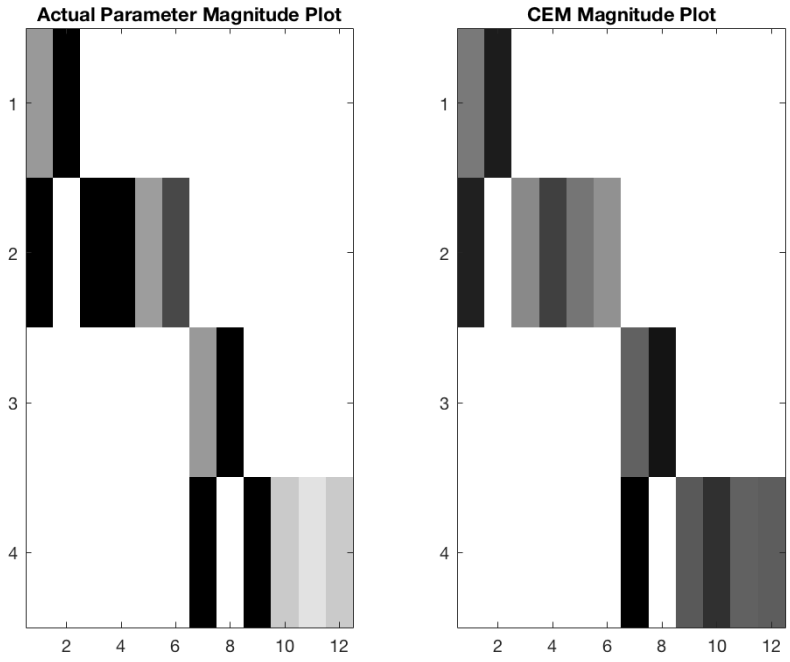

Figure 3.3: Magnitude plots of actual system matrix and estimated CEM for sinusoidal input to Pendulum on a Cart

3.1.3 Projectile Angle of Attack Dynamics Example

The dynamics of a projectile in atmospheric free flight are governed by the six-degree-of-freedom (6DOF) equations of motion described in [59]. These equations govern both the translation and rotational motion of the projectile, which are subject to substantial nonlinear coupling. In general, the rotational motion of the projectile occurs at much faster timescales than changes in velocity, and thus projectile aerodynamic and stability properties are commonly studied by analyzing angular motion in the angle of attack and angle of sideslip, which can be constructed from body-frame velocities. Several key aerodynamic coefficients that determine projectile stability can be identified by studying the angle of attack dynamics in particular. As shown in Fig. 3.4, the angle of attack α (referred to here as $x^{(1)}$) and angle of sideslip β (referred to here as $x^{(3)}$) describe the projectile orientation with respect to its velocity vector. $V_{cg/I}$ refers to the velocity of the projectile center of mass with

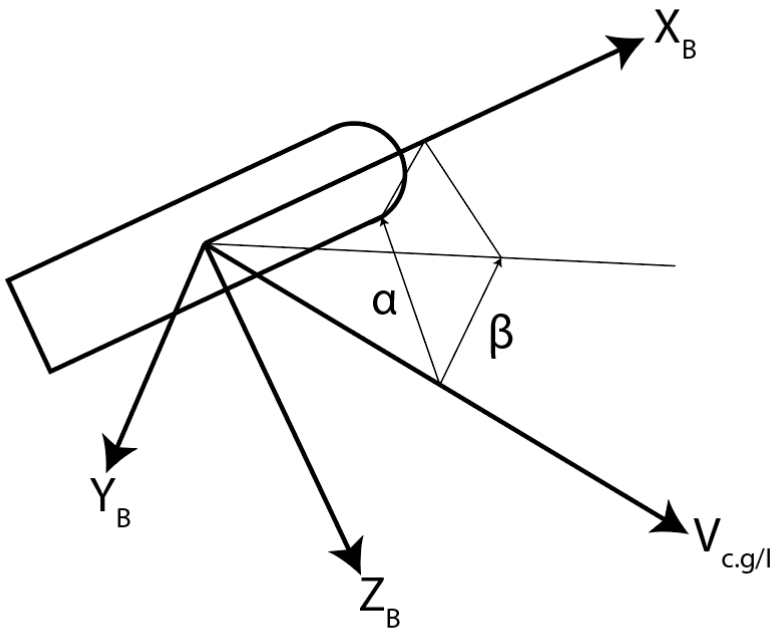

Figure 3.4: Diagram of angle of attack dynamics

respect to an inertial reference frame. The dynamics of these quantities are governed by key aerodynamic parameters, in particular the pitch damping (denoted as M_q) and pitching moment dependence on angle of attack (denoted as M_α). Recently, a reduced-order nonlinear model has been derived describing the angle-of-attack dynamics of a projectile in free flight [60]. These equations of motion were developed to investigate nonlinear stability properties, and allow for analysis of angle of attack dynamics and projectile stability without needing to consider the full 6DOF equations. The equations of motion for this reduced-order system were derived in [60] and are summarized in Equations (3.14)-(3.20):

$$\dot{x}^{(1)} = x^{(2)} \tag{3.14}$$

$$\dot{x}^{(2)} = -x^{(4)^2}\sin(2x^{(1)}) - bx^{(4)}\cos(x^{(1)}) + \frac{M}{I_y} \tag{3.15}$$

$$\dot{x}^{(3)} = x^{(4)} \tag{3.16}$$

$$\dot{x}^{(4)} = \frac{bx^{(2)} + 2x^{(2)}x^{(4)}\sin(x^{(1)}) + \frac{N}{I_y}}{\cos(x^{(1)})} \tag{3.17}$$

$$b = (\dot{\phi} - x^{(4)}\sin(x^{(1)}))\frac{I_x}{I_y} \tag{3.18}$$

$$M = M_\alpha \sin(x^{(1)})\cos(x^{(3)}) + M_q x^{(2)} - N_\alpha \sin(x^{(3)}) \tag{3.19}$$

$$N = M_\alpha \sin(x^{(3)})\cos(x^{(3)}) + M_q x^{(4)}\cos(x^{(1)}) - N_\alpha \sin(x^{(1)})\cos(x^{(3)}) \tag{3.20}$$

In the above equations, $\dot{\phi}$ is the projectile spin rate, I_x and I_y are the axial and transverse moments of inertia respectively, and M_α, M_q, N_α are aerodynamic coefficients of interest that relate to the pitching and yawing dynamics of the projectile [60]. These coefficients are typically estimated from wind tunnel data or flight experiments in a spark range [61] using Maximum Likelihood Estimation. However, for novel or non-traditional configurations it may be difficult to formulate initial guesses for the parameter set, or to identify if any of the parameters are small enough to neglect. The following examples demonstrate the ability of the causation entropy matrix to reveal the relative importance of each of the parameters in shaping the angle of attack response and to highlight which coefficients (if any) are zero or small enough to be neglected.

Equations (3.14)-(3.20) can be discretized as done in previous examples and put into the form of Equation (1.11). The result is a 4×12 matrix Θ, where each element of Θ is a function of the system parameters. The matrix is not shown here for space reasons. Note that several entries of Θ are proportional to M_α or N_α. Thus, by examining these particular matrix entries the relative magnitude of M_α and N_α is revealed. However, it turns out that the damping coefficient M_q is not directly observable using this approach of

discretization. Consider the fact that in Equation (3.15), after substituting the aerodynamic moment expansion in Equation (3.19), the resulting equation becomes,

$$\dot{x}^{(2)} = -x^{(4)^2}\sin(2x^{(1)}) - bx^{(4)}\cos(x^{(1)}) + \frac{M_\alpha \sin(x^{(1)})\cos(x^{(3)}) + M_q x^{(2)} - N_\alpha \sin(x^{(3)})}{I_y}$$
$$= f(x_1, x_2, x_3, x_4) + \frac{M_q x^{(2)}}{I_y} \quad (3.21)$$

After discretizing the system using Equation (2.2), the state update equation for $x^{(2)}$ will contain two terms involving $x^{(2)}$ — one multiplied by M_q, and another constant term. This expansion is shown in Equation (3.22). In (3.22), T is the time step as defined in (2.2).

$$x_{t+1}^{(2)} = T(f(x_t^{(1)}, x_t^{(2)}, x_t^{(3)}, x_t^{(4)}) + \frac{M_q}{I_y}x_t^{(2)}) + x_t^{(2)}$$
$$= T(f(x_t^{(1)}, x_t^{(2)}, x_t^{(3)}, x_t^{(4)})) + (\frac{TM_q}{I_y} + 1)x_t^{(2)} \quad (3.22)$$

As can be seen in Eq. (3.22), the coefficient multiplying $x^{(2)}$ is $M_q + 1$. This means that even if M_q is zero, the entry in Θ will not be zero and thus the causation entropy matrix will not yield a zero value for its corresponding entry. It turns out that the same phenomenon will occur with regard to M_q in the discretized version of Equation (3.17) after substituting in Equation (3.20), which is the only other appearance of this parameter in the discretized equations of motion. Thus, in this case it is not possible to identify a zero value for M_q through use of the CEM.

Using the discretized angle of attack dynamics, a simulation was performed from initial conditions $x_1 = 0.17$ rad, $x_2 = 1$ rad/s, $x_3 = 1.5$ rad, $x_4 = 2$ rad/s with a time step of $T = 0.001$ s. The parameters used for this example are given by: $M_\alpha = 346.2525$ Nm, $N_\alpha = -0.496064$ Nm, and $M_q = 169.6915$ Nm/(rad/s). Figure 3.6 shows the time history of the states for this simulation. The plot displays a variable-timestep ODE solution to the continuous dynamics as well as the forward difference approximation, showing nearly identical trajectories. Figure 3.5 is the magnitude plot for the parameter matrix Θ (left)

and the causation entropy matrix (right) estimated from the time series data. The causation entropy matrix had a 100% (48/48) covariate selection accuracy in identifying zero and nonzero parameters. Furthermore, as illustrated by the correspondence between the color intensities in each plot, the dominant system parameters and secondary system parameters are clearly identified. This knowledge can be used to formulate a more accurate initial guess for a subsequent MLE aerodynamic coefficient estimation process.

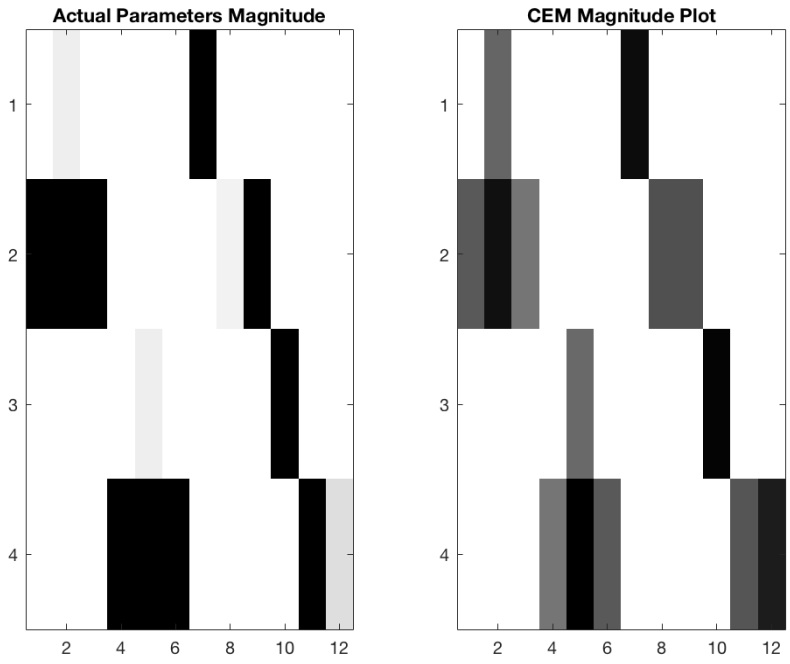

Figure 3.5: Magnitude plot for projectile with all parameters set to nonzero values

The same process as above was repeated using the same parameter values except with $N_\alpha = 0$. Figure 3.7 shows a time history of the states in this case, which exhibit a clear difference with respect to Fig. 3.6. Figure 3.8 shows the magnitudes of the parameter matrix entries and causation entropy matrix corresponding to the time response in Fig. 3.7. The causation entropy matrix had a covariate selection accuracy of 97.9% (47/48) in identifying zero and nonzero parameters. The one instance of error was an incorrect return of a nonzero causation entropy when the parameter was zero; however, the causation entropy reported

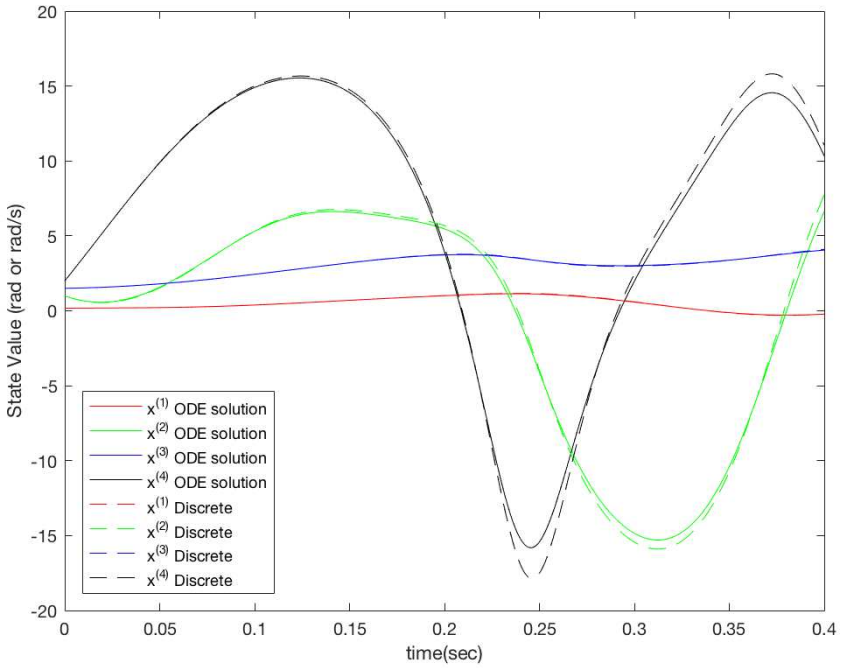

Figure 3.6: Time history of projectile states with nonzero values for all parameters

for this value was three orders of magnitude smaller than all other responses in the entropy matrix. It is important to note that the (2,3) and (4,6) elements of Θ are proportional to N_α so that, if $N_\alpha = 0$, these terms are zero as well. Figure 3.8 shows that these parameters (and CEM values) are in fact zero, unlike in Fig. 3.5. Thus, in an actual parameter estimation scenario, if these elements of the CEM matrix are seen to be zero it can be immediately inferred that $N_\alpha = 0$, or that N_α is at least small enough that it has negligible effects on the dynamics. In this way, particular entries of CEM can be used to determine whether parameters are small enough to be eliminated from the estimated set during MLE.

A series of additional simulation experiments was performed with other combinations of system parameters. These additional experiments are omitted here for space reasons, but the overall conclusion from these results is that the CEM correctly classifies parameters as zero or non-zero better than 95% of the time, and also reveals the relative parameter magnitudes for any combination of M_α, M_q, N_α. However, as mentioned above, information

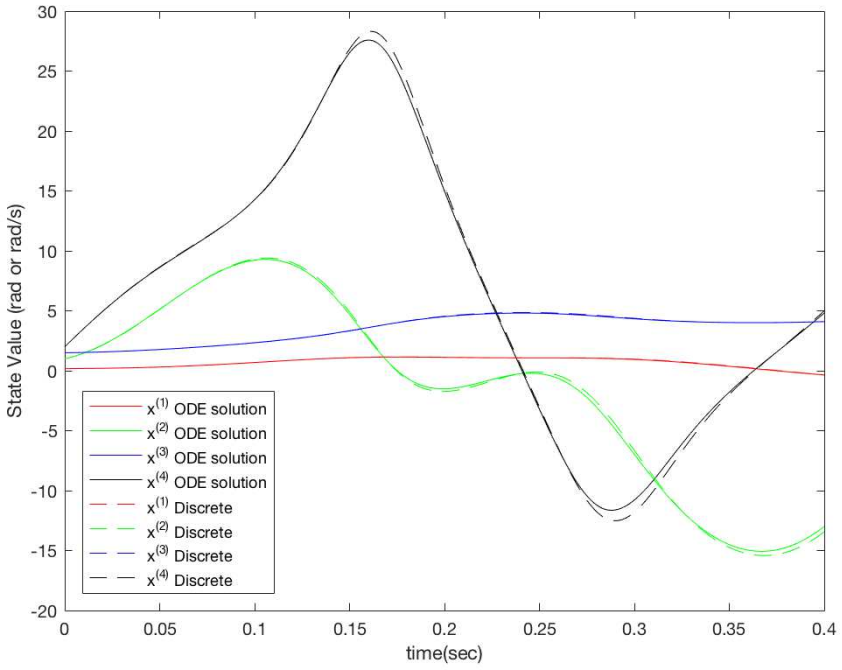

Figure 3.7: Time history of projectile states with $N_\alpha = 0$, $M_\alpha, M_q \neq 0$

regarding the overall magnitude of M_q is difficult to obtain because it is unobservable with respect to additional non-zero damping stemming from gyroscopic terms in the dynamics. Overall, this example represents an important advancement in transitioning use of the CEM to practical physical problems of interest involving nonlinear parameter estimation.

Parameter Number Growth

In Section 3.1.3 the CEM was largely proposed as a method for identifying the necessity of the parameters M_α, M_q and N_α. However, the results shown above demonstrate a CEM with 48 total entries that were calculated, which represents a significant growth in the number of parameters to compute based upon discretization of the system from the 6 parameters that occur relating to M_α, M_q and N_α in the continuous time equations of motion. However, only 6 of the entries in the CEM relate to the parameters in question. The fact that there are 6 instances in the discrete and continuous equations is not a coincidence. In fact,

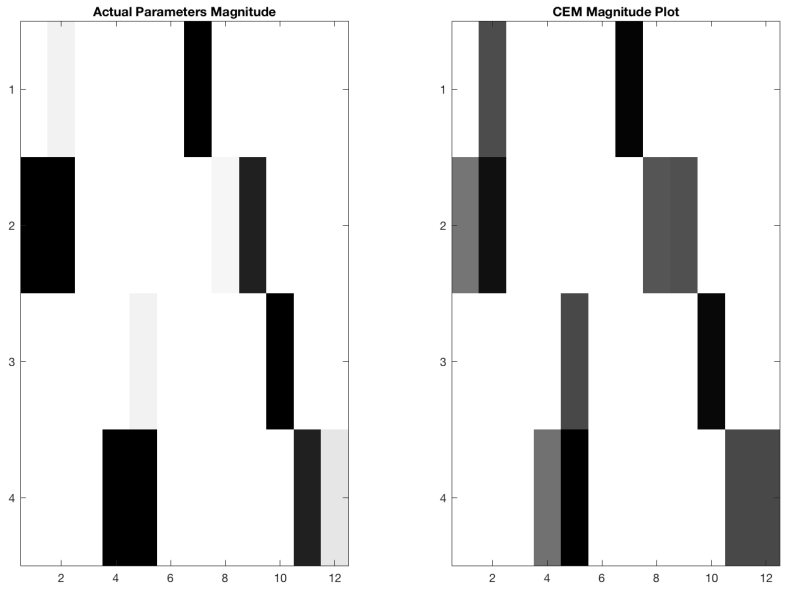

Figure 3.8: Magnitude plot for projectile with $N_\alpha = 0$, $M_\alpha, M_q \neq 0$

it will always be the case that the number of instances of the parameters in question appear in the discrete time equations of motion will always be equal to the number of instances in the continuous time equations. The reasoning is demonstrated below in Equation (3.23) where ρ is the parameter in question and a_i and f_i are the parameters and functions that compose the linearly parameterized remainder of the dynamics.

$$\begin{aligned} \dot{x} &= f(x,\rho) = \rho f_1(x_t) + \sum a_i f_i(x) \\ x_{t+1} &= T\left(\rho f_1(x_t) + \sum a_i f_i(x)\right) + x_t = T\rho f_1(x_t) + T\sum a_i f_i(x_t) + x_t \end{aligned} \tag{3.23}$$

As is visible above, the transformation from Equation (2.2) has no impact on the number of occurrences of a parameter in question and can at most increase the number of total parameters in an equation by 1 (only if x_t is not a function that already exists in the dynamics). Therefore, the discretization of the model will not alter the number of parameters should only some subset of the total continuous time dynamics are considered.

In all examples above, the entire CEM is computed and displayed regardless of any knowledge of the dynamics for completeness. However, practically this is not necessary. All entries in the CEM are causation entropies that can be independently computed without knowledge of any other entries in the CEM. Thus, in the case of Section 3.1.3, computation of the CEM is unnecessary with only the $(2, 1:3)$ and $(4, 4:6)$ entries actually relevant to determine if the parameters in question are necessary in the system model. Computation of only the 6 relevant causation entropies instead of the whole CEM will save both time and a significant computational load; however, as will be demonstrated in Chapter 6, the CEM can sometimes provide unexpected insights into model behavior that can further simplify derived dynamics if the entire CEM is considered.

3.2 Nonzero Causation Entropy Magnitude and Parameter Sensitivity

3.2.1 Sensitivity Overview

Sensitivity analysis is a well-known technique used to provide insight into system parameters and their importance to their respective models. Sensitivity analysis tools are regularly used to quantify the effect of uncertainty on a model parameter [62]. There are many proposed methods for determining the sensitivity of a parameter. In this work when referring to parameter sensitivity, a local, one-at-a time sensitivity measure is used as described by [63, 64]. This method does not capture the coupled sensitivity of system parameters, but it does offer an intuitive and low-order method for understanding how the magnitude of a system parameter effects the overall system response.

Consider a_i to be a parameter in a model for state x with n terms, so that x is defined by Equation (3.24).

$$x_{t+1} = \sum_{i=1}^{n} a_i f(x) \tag{3.24}$$

The notation $x^j_{t+1}(a_0)$ means that x_{t+1} is evaluated with a_j replaced with a_0. Then, the sensitivity of a parameter a_i can be defined as Equation (3.25), where δ is a disturbance

used to perturb the parameter value [62]. For this work, δ is chosen somewhat arbitrarily as 0.15. Intuitively, the sensitivity definition given in Equation (3.25) represents the degree to which an increase in the magnitude of a given parameter will lead to a change in the magnitude of the response of a given state vector component. This metric can be used to create a formalized ranking of the relative importance of each parameter. Selection of a different definition of sensitivity can lead to a slightly different rankings regarding which parameter is most important. However, most methodologies tend to return similar outcomes [62]. Thus, for this study, the definition of sensitivity provided in Eq. (3.25) is used.

$$sens(a_i) = x^i_{t+1}(a_i(1+\delta)) - x^i_{t+1}(a_i) \tag{3.25}$$

This notion of sensitivity is rather logical as an importance metric, as an entry with high sensitivity will have a higher effect on the error based on any relatively sized perturbation of the parameter, thus including it accurately is of the utmost importance.

3.2.2 Sensitivity and Causation Entropy Magnitude

For this section, a set of coupled linear mass-spring-dampers as well as the inverted pendulum from Section 3.1.2 are considered. The equations of motion for the coupled mass-spring-damper system mass j are given by Equation (3.26). When applying (3.26) to a mass at the edge, any term that appears in (3.26) that does not apply is replaced with a zero.

$$\ddot{x}_j = \frac{1}{m_j}[(K_{j+1}(x_{j+1} - x_j) - K_j(x_j - x_{j-1}) + C_{j+1}(\dot{x}_{j+1} - \dot{x}_j) - C_j(\dot{x}_j - \dot{x}_{j-1})] \tag{3.26}$$

Values for the parameters of the coupled mass-spring-damper system in all simulations were chosen as follows: $m_1 = m_2 = m_3 = m_4 = 1$ kg, $c_1 = 0.1$ kg/s, $c_2 = 0.11$ kg/s, $c_3 = 0.12$ kg/s, $c_4 = 0.13$ kg/s, and $K_1 = 4$ N/m, $K_2 = 5$ N/m, $K_3 = 6$ N/m, $K_4 = 7$ N/m. All simulations of the pendulum on a cart system used parameter values identical to

in Section 3.1.2.

Simulations of both systems were run with the CEM computed for the trajectory as well as the sensitivity of each parameter. For comparisons in this section, a parameter's importance is its ranking in terms of magnitude of the causation entropy or sensitivity with a ranking of one meaning the parameter had the lowest causation entropy/sensitivity seen and the parameter with the highest causation entropy/sensitivity having the highest. Figures 3.9 and 3.10 demonstrate the strong correlation between the parameter sensitivity and the causation entropy importance. Each plot demonstrates the parameter's ranking in terms of causation entropy on the left and sensitivity on the right. It is clear that the relative magnitude of the causation entropy provides a strong insight for the relative sensitivity of the parameter. Thus, the larger the nonzero magnitude of the causation entropy matrix, the larger the parameters sensitivity will likely be as compared to other sensitivities in the system, which suggests a greater importance of the parameter. Thus, the causation entropy magnitude provides a proxy for understanding the relative sensitivity of the corresponding parameter. Greater discussion of the relationship of a parameter sensitivity and the causation entropy magnitude particularly in the presence of noise is presented in Section 5.1.2.

The ability of the CEM to identify the relative sensitivities of the parameters has benefits beyond the ability to identify parameters as will be demonstrated in Chapter 4. Knowledge of the sensitivity allows for more informed decision making on where to focus efforts on identifying parameters accurately in order to have an accurate model. Usually, sensitivity is estimated through either attempting to estimate the local gradients of the parameters or using the experimental setup to physically change the parameters and measure the change in output [65]. This requires either access to the system, which can be impossible or very expensive, or the ability to estimate gradients which can require significant amounts of data to perform accurately, which may be unavailable. The CEM allows for accurate estimation of the parameters sensitivity from one, potentially short, set of data.

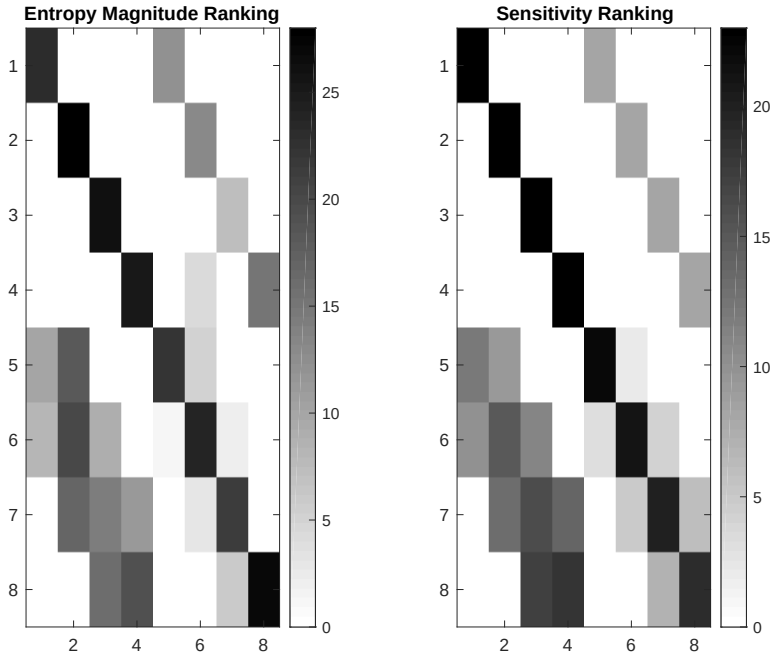

Figure 3.9: MSD sensitivity vs causation entropy ranking

Figures 3.9 and 3.10 demonstrate that the relative magnitude of the causation entropy magnitude is proportional to the relative sensitivity of a parameter. Thus, generally the the causation entropy magnitude for a given parameter, the greater impact exclusion or error of said parameter will cause. Thus, parameters with small causation entropies (especially relative to other causation entropies seen) will have a minimal overall impact on overall model output and thus likely performance. This phenomenon is further explored in Chapter 4.

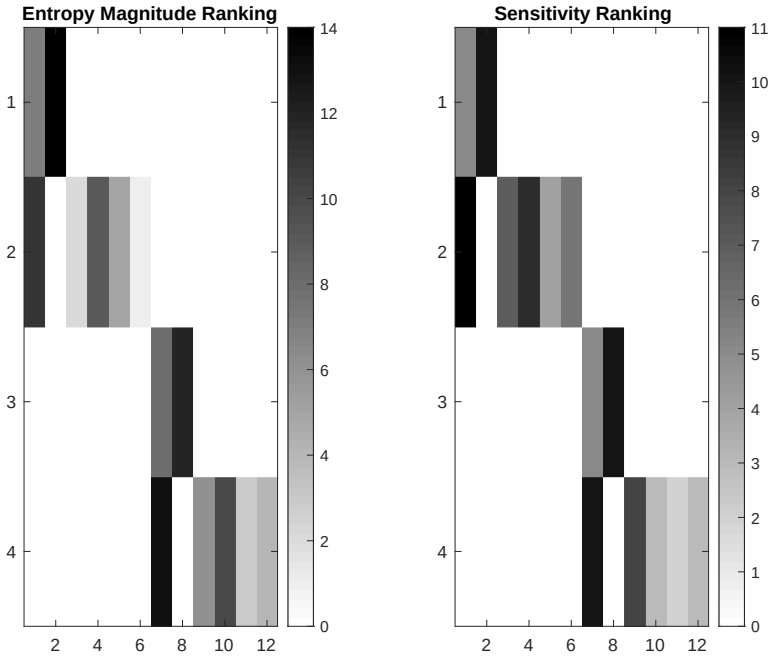

Figure 3.10: Inverted Pendulum sensitivity vs causation entropy ranking

CHAPTER 4
APPLICATION OF THE CEM TO BLACK BOX MODELS

This chapter explores the application of the CEM to black box systems. Section 4.1 provides background on black box models with section 4.1.3 demonstrating the application of a black box model to a nonlinear car suspension problem, which highlights not only the ability to identify unnecessary parameters in a black box problem, but also the benefits of knowledge of the CEM magnitudes and their relationship to parameter sensitivity. The section concludes with a comparison of the CEM performance with current state of the art covariate selection and optimization techniques.

4.1 Black Box Models

4.1.1 Black-Box Model Background

Black box system identification is employed in cases where there is either no *a priori* knowledge about the system in question, or when a low order model is used as a surrogate to represent complex dynamics. Unlike Section 3.1 where the CEM was applied to parameterized models derived from physical laws (grey box models), this section considers cases in which black box models are fit to experimental data. The ultimate goal of this fitting process is to generate a model that can predict the dynamics of the system in question. One class of black box identification methods seeks to approximate the system dynamics using polynomials [5]. Recorded time series data for the response of a system is compared to the output of the approximated polynomial model. The polynomial coefficients are then optimized through a nonlinear optimization process defined over a least squares error cost between the predicted and actual states.

There are two decisions that must be made when constructing a polynomial model

approximation: the underlying structure of the polynomial and the order of the polynomial. Many different structures for polynomial models have been proposed, and the problem of choosing the polynomial structure for a given problem is beyond the scope of this work. Rather, this thesis will consider only a subclass of the Kolmogorov-Gabor polynomials, namely, the nonlinear differential equation (NDE) model described in [5]. This model is studied here primarily because it is structurally similar to many mechanical systems, and thus can be used to form a reasonable approximation of their dynamics. Extension of the proposed CEM methodology to other polynomial model structures is straightforward but is not investigated here.

When creating a polynomial model approximation, the goal is to be able to predict the dynamic response so that the model not only fits the training data, but will also generalize to data that was not used to train the model. This requires that the polynomial be of a high enough order to be able to represent the dynamics. It appears tempting to create a polynomial of the highest possible order; however, this causes two issues. The first is that a high order polynomial can be computationally expensive for the parameter optimization routine. The second is that overfitting of the data can occur when running the nonlinear optimization due to high dimensionality. Overfitting can occur when a polynomial with significantly more terms than needed is used. When the polynomial order is too high, there are potentially multiple, non-unique sets of coefficients that lead to similar trajectories, which implies that multiple local minima are present in the cost function surface. From a practical standpoint, overfitting is problematic because, although the model can represent the training data well, it is not predictive in the sense that it will not accurately predict the model response to conditions outside the training data set. As will be demonstrated, this phenomenon can be avoided by finding a minimal-order polynominal representation of the model. This can be accomplished by applying the CEM to the measured state time histories as a pre-processing step prior to coefficient optimization. Once the CEM is computed, the non-zero coefficients can be identified and a reduced parameter set consisting only of these

coefficients can be optimized. This process is demonstrated in this section through a series of examples.

For the remainder of this section on Black-Box Models, a new notation is adopted to allow for clearer representation of polynomial models. For continuous time systems, the notation x_i refers to the i^{th} state of x. For discrete time systems, the notation of $x_{i,j}$ represents the i^{th} state of x at timestep j.

4.1.2 NDE Model Structure

NDE models are a class of polynomial-based black box models used in system identification. The dynamic order of the NDE model is the number of previous time steps that are used in the polynomial approximation of the system. In order to match the formulation of the CEM, only NDE models with dynamic order 1 are considered. In general, to maintain consistency between the causation entropy definition and the polynomial approximation, the dynamic order of the NDE model should match the selected value of τ_x, τ_y, etc.

To introduce the NDE model structure, consider a three-state system with state vector $\{x_1, x_2, x_3\}^T$. Using a third-order NDE model, the state update equation for x_1 is given by [5],

$$\begin{aligned} x_{1,t+1} = \theta_1 u_t + \theta_2 x_{1,t} + \theta_3 x_{2,t} + \theta_4 x_{3,t} + \theta_5 x_{1,t}^2 + \theta_6 x_{2,t}^2 + \theta_7 x_{3,t}^2 + \theta_8 x_{1,t}^3 + \theta_9 x_{2,t}^3 ... \\ + \theta_{10} x_{3,t}^3 + \theta_{11} x_{1,t} * x_{2,t} + \theta_{12} x_{1,t} * x_{3,t} + \theta_{13} x_{2,t} * x_{3,t} + \theta_{14} x_{1,t} * x_{2,t} * x_{3,t} + \theta_{15} x_{1,t}^2 * x_{2,t} + ... \\ \theta_{16} x_{1,t}^2 * x_{3,t} + \theta_{17} x_{2,t}^2 * x_{1,t} + \theta_{18} x_{2,t}^2 * x_{3,t} + \theta_{19} x_{3,t}^2 * x_{1,t} + \theta_{20} x_{3,t}^2 * x_{2,t} \end{aligned} \quad (4.1)$$

In (4.1), each θ_i is an unknown parameter that must be optimized based on observed data. Analogous, but separate, NDE models can be constructed for $x_{2,t+1}$ and $x_{3,t+1}$. In the work presented below, the gradient-based Levenberg-Marquardt algorithm was used for parameter optimization [66]. It is worth noting that even for a system with three state variables, a third order polynomial will have 20 parameters that must be optimized per state

equation implying 60 parameters are needed to generate the systems' equations of motion. The number of parameters n_p in an NDE polynomial with dynamic order d is given by (4.2) where r is the order of the polynomial and s is the number of states in the system. This equation is derived from the number of generalized combinations with repetition as provided in [67].

$$n_p = \sum_{i=1}^{r} \frac{(d \times s + i - 1)!}{i!\,(d \times s - 1)!} \tag{4.2}$$

Thus, even when considering dynamic order one models only, as the size of the state space grows the size of the parameter space increases quickly. The examples below demonstrate the advantages of using the CEM pre-processing technique to reduce the dimensionality of the parameter space for this type of model.

4.1.3 Quarter Car Suspension with Nonlinear Stiffness

A black-box example considers the case of a car suspension model. Figure 4.1 represents a common model of a quarter car suspension [68], where each spring has nonlinear response characteristics. In this example, the system response to a sinusoidal input $u = 0.1\sin(t)$ is considered. Each spring uses the same model as in Eqn. (4.3), where β_1 and β_2 are the cubic response coefficients of springs 1 and 2 respectively.

$$f_s(x) = kx + \beta x^3 \tag{4.3}$$

Suppose data from a given car suspension design is collected experimentally, and a model is to be constructed. It is typically unknown *a priori* whether the springs and dampers should be modeled linearly, or if nonlinear terms should be included. If the spring is in fact linear, but a nonlinear model is fit to the data, overfitting issues as demonstrated previously could occur, while if a linear model is fit to a nonlinear system, the model will be of poor fidelity and will provide inaccurate predictions. Use of the CEM pre-processing method will allow for selection of an appropriately complex model that avoids overfitting but also

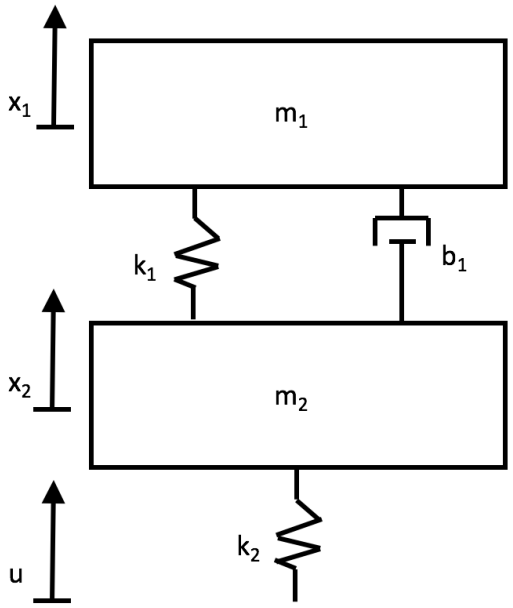

Figure 4.1: Quarter car suspension model

provides favorable fidelity.

For this system, the equations of motion can be represented as,

$$\dot{y}_1 = y_2 \tag{4.4}$$

$$\dot{y}_2 = (1/m_1)[-b_1 * (y_2 - y_4) - k_1 * (y_1 - y_3) - \beta_1 * (y_1 - y_3)^3 \tag{4.5}$$

$$\dot{y}_3 = y_4 \tag{4.6}$$

$$\begin{aligned}\dot{y}_4 = (1/m_2)[b_1 * (y_2 - y_4) + k_1 * (y_1 - y_3) + \beta_1 * (y_1 - y_3)^3 - \dots \\ k_2 * (y_4 - u) - \beta_2 * (y_4 - u)^3]\end{aligned} \tag{4.7}$$

For all cases in this section the model parameters are selected as: $m_1 = 12$ kg, $m_2 = 7$ kg, $k_1 = 10$ N/m, $\beta_1 = 23.5$ N/m^3 , $b_1 = 0.075$ N-s/m, $k_2 = 8$ N/m, $\beta_2 = 14.25$ N/m^3, and $T = 0.01$ s. A third order NDE model is chosen as a black box representation for the velocity dynamics of mass one (y_2). Given the NDE model form in (4.1), this model has 35 total parameters per state equation in its general form. From Equation (4.5) it is clear

that each term in the actual dynamics is a polynomial of the state variables, and thus the actual dynamics can be represented exactly by the NDE model. However, many of the 35 parameters are extraneous, and their use can potentially lead to overfitting of the model similar to previous examples.

An example trajectory was generated with this system using the forcing input described above. The CSE was then computed from this example trajectory and the permutation test applied, resulting in the magnitude plot shown in Figure 4.2. From these results, it is clear that only parameters $1, 2, 3, 4, 9, 11, 24$ and 29 provide any information to the model. This can be easily verified by expanding the discrete approximation to (4.5) to reveal which terms of the NDE model match the exact dynamics. Parameter 4 provides significantly less information (0.19 nats) compared to the other terms, which range from 2.66 nats to 6.13 nats. Thus, depending on the use and required accuracy of the model, the fourth term could be included or omitted without significant loss of accuracy of the optimized model. For the sake of illustration, results both including and omitting the fourth parameter from the reduced set are presented here. In order to test this, parameter sets were optimized for the complete parameter set, the CEM reduced parameter set with parameter 4 and the CEM reduced parameter set without parameter 4. 500 random initial conditions were then generated with each of the models forward propagated from each of the initial conditions and the average mean squared error (MSE) between the forward propagated dynamics and true dynamics computed and recorded. For all cases, state equations for states 1, 3 and 4 were generated by the idealized dynamics. If the MSE for a given propagation exceeded 50,000, it was considered to be unstable and have diverged. In this case, the model performance was discarded. The average MSE was then taken from only the available MSE values, but the number of cases of having an unstable propagation is reported. The MSE results for the full- and reduced-order models when compared to the corresponding idealized trajectory are provided in Table 4.1. Initial guesses for the parameters were drawn from the same distribution (0 mean normal distribution with a standard deviation of 0.05) with guesses

Table 4.1: Comparison of error metrics for full- and reduced-order NDE models

Model Error Comparison			
	Training Error	Monte Carlo MSE	Numb. Unstable Prop.
Full Param. Set	9.7545×10^{-05}	0.1424	49 (9.8%)
Reduced Param. Set w/ Param. 4	1.0559×10^{-21}	7.6213×10^{-19}	0 (0%)
Reduced Param. Set w/o Param. 4	7.2233×10^{-08}	3.1423×10^{-04}	0 (0%)

corresponding to equivalent parameters chosen to be identical.

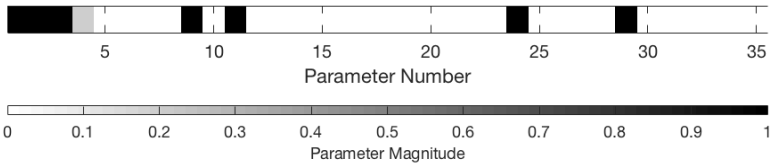

Figure 4.2: Magnitude plot for the CE values for suspension model example

In Table 4.1, the trajectory error is the training error or the error that occurs between the optimized model and the data that it was trained on. Based on the very small values for the full order and reduced order trajectory errors, both sets of models were able to identify a satisfactory model over the training data. Figure 4.3 demonstrates the model fit over the training data of the optimized models. The blue data set is difficult to see as it is almost directly beneath the black line. Clearly, all of the models were able to very accurately approximate the true training data (the red trajectory).

The Monte Carlo MSE represents the average MSE for the cases when the model was propagated on never before seen initial conditions. As shown in Table 4.1, the average

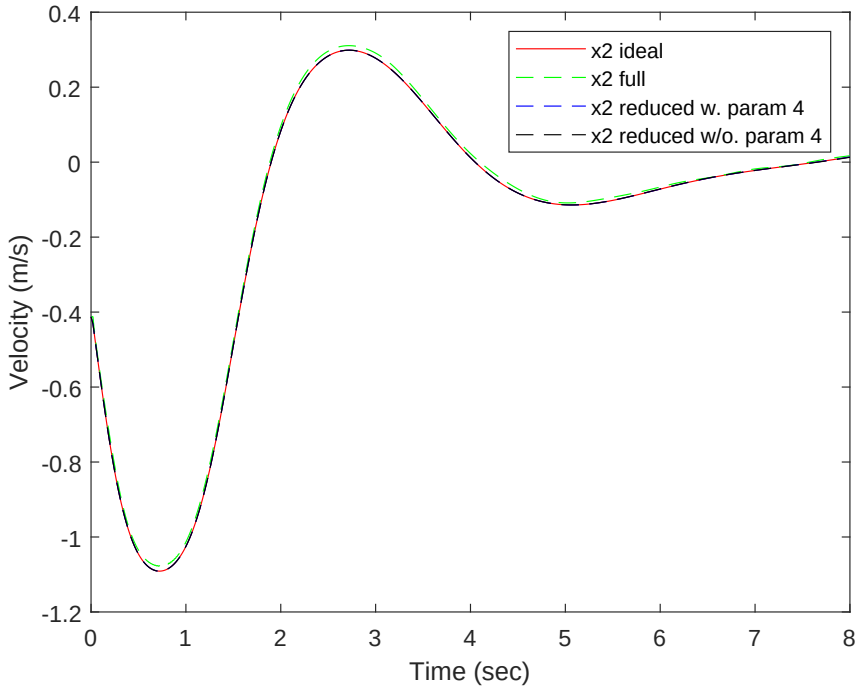

Figure 4.3: Forward propagated optimized models using initial conditions from training trajectory compared with ideal trajectory

MSE for a 500-case Monte Carlo simulation is significantly higher for the full-order model, compared to the reduced-order models (which, in fact, had essentially negligible error). Thus, the full-order model suffers from overfitting. The excess parameters used in the full parameter set provide a close data fit on the training data in Table 4.1; however, when presented with data that deviates from the training data, the overfit parameters can lead to a significant degradation in model performance as demonstrated by the significant increase in average MSE for the Monte Carlo study. An example of the results of propagation for each of the models on a set of identical, randomized initial conditions is provided in Figure 4.4. Beyond the significant reduction in predictive accuracy, the additional parameters can cause the model to become unstable when used on a different region of data. Nearly 10% of propagated cases become unstable (i.e have an MSE larger than 50,000) with the full parameter set while both reduced cases have zero instances of the model becoming unstable or diverging. The reduced-order models both with and without parameter 4 show

much better predictive behavior as the true model is identified and thus any issues with overfitting are avoided.

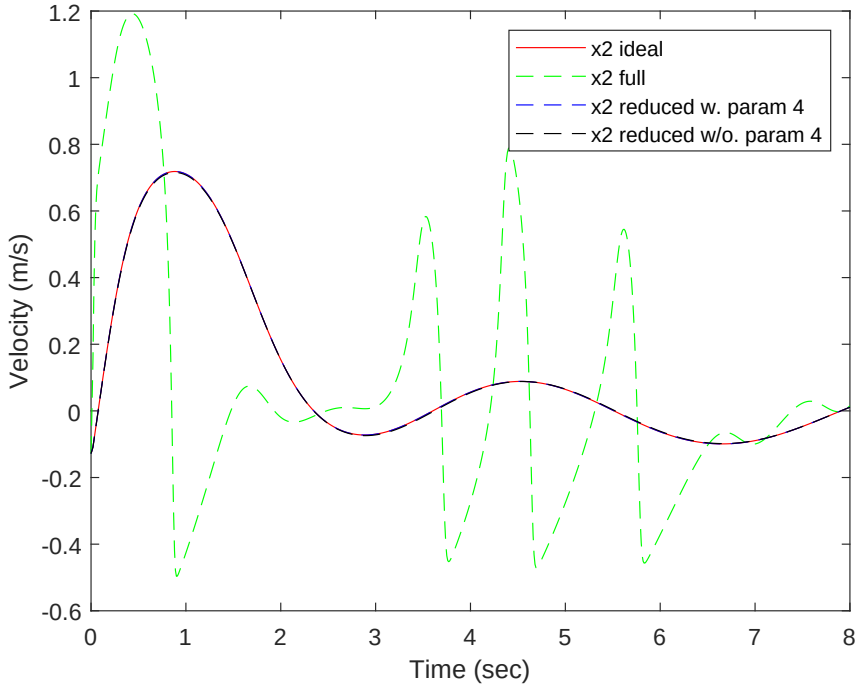

Figure 4.4: Forward propagated optimized models using new initial conditions compared with ideal trajectory

Consider the case where parameter 4 is omitted from the reduced-order model. The MSE for the training trajectory are given in Table 4.1, showing that the model provides an adequate fit (as shown in Figure 4.3). The average MSEs for the Monte Carlo simulation, also shown in Table 4.1, again demonstrates that the reduced-order model exhibits much better predictive behavior than the full-order model, even when excluding parameter 4. When comparing the case of the reduced sets with and without parameter 4, excluding it resulted in a higher average MSE. This is explained by the fact that this term is needed in the NDE model to match the actual dynamic equations, but the causation entropy magnitude suggests that it provides only a relatively small amount of information to the state update. The average MSE for the reduced set without parameter 4 is still orders of magnitude better than the full parameter set. As Figure 4.4, the reduced parameter sets can provide a vast

improvement on the predictive accuracy as compared to attempting to optimize the entire parameter set when used on never before seen data due to the ability to limit the effects of overfitting.

The result on removal of parameter 4 demonstrated above is predictable and in line with the sensitivity discussion included in Chapter 3. The relatively small magnitude of CEM entry corresponding to parameter 4 suggests that the parameter's sensitivity is low, which means that errors in the parameters magnitude will have a minimal effect on the model's overall output. By extension, this includes altering or mismatching of the parameters value by changing it to zero.

4.2 State-of-the-Art Sparsity Identification Techniques

The CEM has been proposed as a methodology to identify zero entries in a parameter set to accurately identify the true model structure of the generative dynamics to reduce both model complexity and chances of numerical overfitting. A natural question that arises, which this section seeks to answer, is how the capabilities of the CEM compare to current state of the art techniques for sparse model fitting. This section compares the shrinkage estimator techniques of LASSO and Elastic Net to the CEM in both covariate selection accuracy and overall performance of optimized models.

4.2.1 Mathematical Formulations of LASSO and Elastic Net Algorithms

Shrinkage Methods Background

Estimating a system model actually involves two separate, but related, problems: selection of the variables or functions that should be included in the model, and, once this decision is made, estimation of the model parameters themselves. The first of these tasks is sometimes referred to as covariate selection or feature selection. This step is important because the inclusion of too many covariates in a model may result in model overfitting or an increased tendency of a parameter optimization technique to converge to local minima as demon-

strated in the previous section [7, 15, 16]. The problem of overfitting and convergence to local minima is particularly problematic in cases when limited training data is available or when data is subject to disturbances or noise [10].

Many common shrinkage estimators attempt to improve regression results by exploiting the so-called bias-variance tradeoff: some amount of bias in model predictions is accepted in order to achieve a lower variance [24]. The least absolute shrinkage and selection operator (LASSO) [25, 26] and elastic net [27] regression methods are two common techniques that improve regression results by eliminating certain model covariates. However, like other shrinkage estimators, LASSO and elastic net require the tuning of hyperparameters that affect how much shrinkage actually occurs. Improper tuning of the hyperparameters may result in not enough terms being removed (potentially leading to overfitting) or too many terms being removed (leading to poor predictive performance). Thus, the tuning of these hyperparameters is an additional task that must be performed which has a significant effect on the degree of model improvement.

LASSO LASSO optimization uses an L_1 norm as a penalty in the optimization routine. The formulation of LASSO was originally presented in [25] with the problem formulated as given in Equation (4.8). The goal is to find the regressor set β that minimizes Equation (4.8).

$$\min_{\beta} \left(\sum_{i=1}^{n} (Y_i - \sum_{j} \beta_j X_{i,j})^2 \right) \quad s.t. \quad \sum_{j} |\beta_j| \leq k \tag{4.8}$$

In Equation (4.8) k is a tuning parameter that must be greater than or equal to 0. This formulation is frequently written in the more familiar Lagrangian form as in Equation (4.9).

$$\min_{\beta} \left(\frac{1}{2} ||Y - X\beta||_2^2 + \lambda ||\beta||_1 \right) \tag{4.9}$$

In Equation (4.9), the Lagrange multiplier λ is a scalar hyperparameter defined greater than or equal to 0 that must be tuned. In Equation (4.9), $||\cdot||_p$ is the standard L_p norm.

Geometric Interpretation Returning to the LASSO in formulation in Equation (4.8), the portion before the $s.t.$ is the objective function and the remainder the constraint. The objective function is the same as that used in least squares or ridge regression. Ridge regression [69] is like LASSO but uses a 2-norm penalty like in the objective function instead. Figure 4.5 demonstrates how LASSO encourages a sparse result in two dimensions. The parameter set selected will occur where the constraint set for a selected value of λ intersects the outermost or highest valued objective function possible. Based on the pointed corners that occur from the 1-norm constraint, there is an increased chance of a sparse result whereas Ridge Regression will always return a fully populated parameter set.

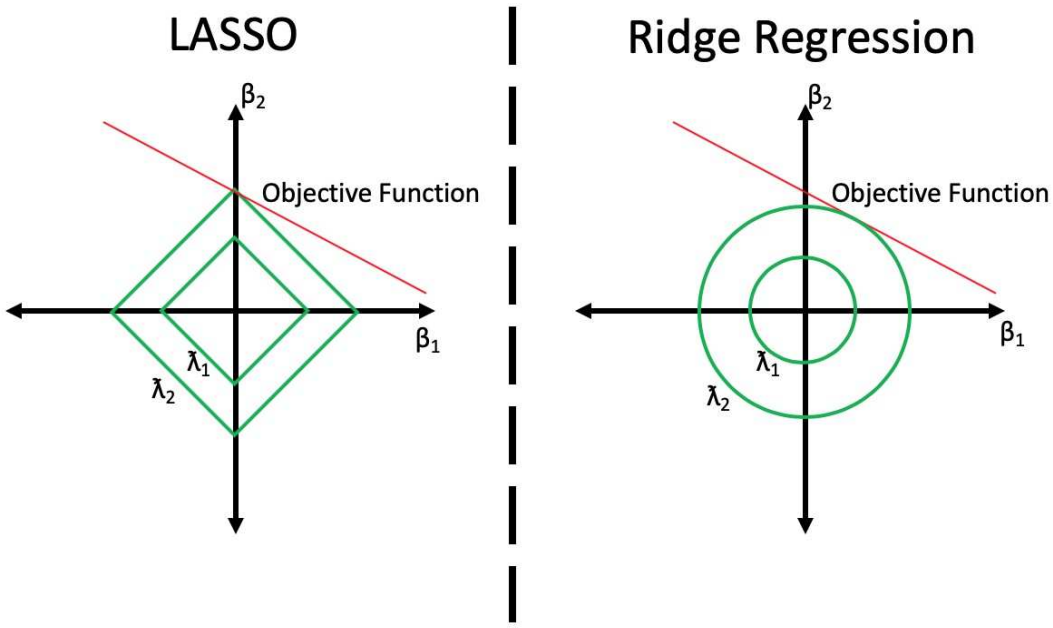

Figure 4.5: Comparison of LASSO and Ridge Regression parameter space

LASSO Implementation In order to effectively use the LASSO technique, λ must be chosen in an intelligent way. The goal of this work is to compare the regularization techniques and the effectiveness of the CEM at identifying a model's sparsity structure. The CEM is largely intended for use in cases with limited data available. Thus, a 1-fold cross validation was used to ensure that groups have sufficient data within to be able to fit

the parameters. A 75-25 split was used between training and testing/validation groups. In order to select λ a set of potential λ values was generated on a grid of 1000 equally spaced points on a logarithmic scale from 10^{-5} to 10^{-1}. For every λ, a model was fit using the training data by the Python sklearn implementation of the LASSO algorithm. The mean squared error (MSE) was then computed between the predicted output of the validation set when using the model verses what the actual output was. The λ corresponding to the lowest MSE was then selected as the proper λ and the model selected.

Elastic Net Elastic net provides a linear combination of L_1 (LASSO type) and L_2 (Ridge Regression) penalties as shown in Equation (4.10). Elastic net seeks to solve some of the shortcomings of LASSO optimization

$$\min_{\beta} ||Y - \beta X||_2^2 + \lambda_2 ||\beta||_2^2 + \lambda_1 ||\beta||_1 \tag{4.10}$$

In the Elastic Net formulation the L_1 penalty serves the same purpose in the LASSO formulation to promote sparse results. However, the L_2 penalty corrects potential issues with the LASSO technique. Particularly, in the case where the number of predictors is larger than the number of data points , $p > n$, LASSO will select at most n predictors as it saturates. Additionally, LASSO tends to only select one predictor from a group of predictors with high correlation. The L_2 term solves both of these issues as well as increases the stability of the L_1 regularization path [27]. The border for the constraint region for elastic net is in between the square LASSO constraint and the round Ridge constraint as shown in Figure 4.6 [27] for proper selection of λ_1 and λ_2. Thus, Elastic Net can provide significant benefit over LASSO; however, Elastic Net has a second hyperparamter that must be selected independently in order to use the technique.

Elastic Net Implementation Similar to the implementation of the LASSO algorithm, the Python sklearn toolbox contains an implementation of the elastic net algorightm. The

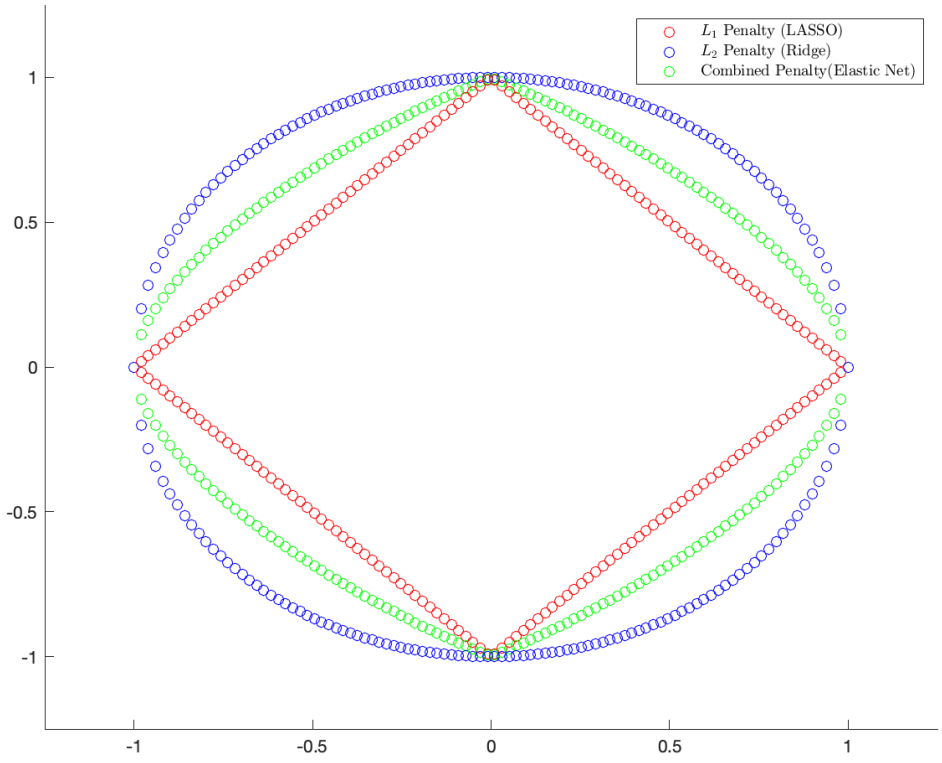

Figure 4.6: Constraint comparison L_1, L_2 and combined $L_1 + L_2$

Elastic Net implementation reparameterizes the hyperparamters to α and an L_1 ratio. The transformations are given in Equations (4.11-4.12).

$$\alpha = \lambda_1 + \lambda_2 \tag{4.11}$$

$$L_1\ ratio = \frac{\lambda_1}{\lambda_1 + \lambda_2} \tag{4.12}$$

The L_1 ratio is defined such that $0 \leq L_1\ Ratio \leq 1$. Note that $\alpha = 0$ corresponds to normal least squares, $L_1\ ratio = 0$ is ridge regression and $L_1\ ratio = 1$ is LASSO.

In order to tune the two hyperparamters a mesh of each parameters was created with scaling as recommended by the sklearn documentation. α was generated as 250 equally spaced points on a logarithmic scale between 10^{-4} and 10^{-1}. The $L_1\ ratio$ was generated as 20 equally spaced points on a linear scale between 0.05 and 1. The same training and

validation scheme as used for the LASSO models was used where a model was trained on the training data and the MSE computed on the validation data with the minimal MSE corresponding to the correct parameter choices.

Mathematical Models of Systems Studied

All simulations used in this work are generated from the discrete model with a time step of $T = 0.01s$.

Linear System A discrete time, linear system is given by Equation (4.13).

$$x_{t+1} = Ax_t \tag{4.13}$$

x is an $n \times 1$ vector and A is an $n \times n$ matrix. Note that this is equivalent to a continuous system after discretization through a first-order finite difference approximation to the derivative as described in [56]. In this case A was defined as in Equation (4.14).

$$A = \begin{bmatrix} 0.99 & 0 & 0 & -0.054 & -0.0135 \\ -0.01 & .9875 & 0 & 0.0075 & 0 \\ 0 & -0.038 & 0.991 & 0 & 0 \\ 0.011 & -0.042 & 0 & 0.98 & 0 \\ 0 & 0 & 0 & -0.011 & 0.9780 \end{bmatrix} \tag{4.14}$$

Van der Pol Oscillator The Van der Pol Oscillator is a well known nonlinear oscillator with the continuous time equations provided below in Equation (4.15)[70].

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \mu x_2 - x_1 - \mu x_2 * x_1^2 \end{aligned} \tag{4.15}$$

For simulations $\mu = 1.15$ was used. In order to test the performance of the model shrinkage techniques, a modified Van der Pol oscillator model is used in which the candidate state function vector $\mathbf{F}$ is augmented with extraneous functions x_1x_2, $x_1x_2^2$, x_1^2 and x_2^2. The oscillator equations are transformed to discrete-time using the transformation technique described in [56] with a time step of $T = 0.01$ sec, yielding a model of the form shown in (1.11). When discretized and written in the form of Equation (1.11), the standard Van der Pol model has 6 total parameters of which 5 are nonzero and only one is zero. This discrete model is shown in Equation (4.16), where $[\cdot]_t$ denotes that the vector elements are evaluated at timestep t. The modified model has 14 total parameters when discretized, of which the same corresponding 5 parameters are nonzero and the remaining 9 parameters are equal to zero. This modified model is shown in discrete form in Equation (4.17). In the results shown below, the modified model in Equation (4.17) is used to generate data, and the goal of the shrinkage estimators is to identify the 9 zero parameters in Equation (4.17) for removal prior to parameter optimization.

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{t+1} = \begin{bmatrix} 1 & T & 0 \\ -T & (1.15T+1) & -1.15T \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_2x_1^2 \end{bmatrix}_t \tag{4.16}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{t+1} = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 & 0 \\ -T & (1.15T+1) & -1.15T & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_2x_1^2 \\ x_1x_2 \\ x_1x_2^2 \\ x_1^2 \\ x_2^2 \end{bmatrix}_t \tag{4.17}$$

4.2.2 Numerical Results of Shrinkage Techniques for Model Optimization

This section describes two numerical studies of model shrinkage performance comparing the CEM technique to LASSO and elastic net with respect to noise level and training data length.

Data Size Study

In this study, the two systems in Eqs. (4.14) and (4.17) were simulated for a fixed amount of time from randomized initial conditions for 50 trials. For each trial, the state time history is processed by the CEM-based algorithm, LASSO, and elastic net to perform covariate selection. Each algorithm is provided the candidate model – in the case of the linear system, the fully-populated A matrix in Equation (4.14), and in the case of the Van der Pol oscillator, the augmented model which includes the extraneous terms. The algorithms are then tasked to select the optimal set of covariates. This process is repeated for various lengths of data sets generated by the dynamics (i.e. simulation duration).

Algorithm performance can be quantitatively evaluated since the ”true” model is known with respect to the candidate model. In the linear case, the candidate model has 25 possible parameters (relating covariates to state updates); however, only 13 are nonzero and thus optimal performance of the shrinkage algorithms should eliminate the 12 zero parameters. Likewise, the candidate Van der Pol model with the added terms has 14 total parameters, of which 9 should be set to zero. Thus, for each simulation trial the *covariate selection accuracy* can be calculated as the number of parameters that the algorithm correctly identifies to be nonzero plus the number correctly identified as zero, divided by the total number of entries in the parameter matrix Θ.

Figures 4.7 and 4.8 show the results of this study for the linear system and Van der Pol oscillator, respectively. The x-axis of each plot shows the data size used in training, while the y-axis shows the covariate selection accuracy (where 1 represents perfect accuracy). For each data size, 50 trials were performed, with the mean, standard deviation, and the

minimum and maximum of the selection accuracy plotted. Interestingly, the results show that the CEM technique exhibits a higher mean selection accuracy (near 1) than LASSO or elastic net which is independent of data size. This is in contrast to LASSO and elastic net, which in the case of the Van der Pol oscillator exhibit notable dependence of accuracy on data length. Additionally, the variance in CEM performance is much less than the variance in performance of LASSO and elastic net, particularly in the Van der Pol oscillator example. In Fig. 4.8, the CEM accuracy fluctuates only between about 80% and 100% for data lengths longer than 50 points, whereas for LASSO and elastic net the performance is highly dependent on the specific time series. For instance, the LASSO accuracy can vary anywhere between 38% and 96% for the 550 data point case, depending on the actual data sequence provided. Overall, these results show that, at least in the case of zero measurement noise, the CEM-based selection method far outperforms LASSO and elastic net in terms of a higher mean accuracy and a lower variation when presented with different measurement sequences. The CEM-based technique also appears to be more accurate with smaller data sizes. Finally, the CEM-based method realizes these advantages while also eliminating the need for tuning of hyperparameters as required by LASSO and elastic net.

Noise and Predictive Accuracy

A second study is conducted to examine the performance of the CEM estimator in the presence of noise compared to that of LASSO and elastic net, as well as the predictive performance of the resulting models. This study focuses on the expanded Van der Pol system in (4.17). Monte Carlo simulations are performed in which zero mean Gaussian measurement noise is added to the time series data after it is generated from random initial conditions. For each trajectory, the random initial conditions for each state are sampled from a zero-mean Gaussian distribution with standard deviation of 2. For the CEM case, CEM optimization was used to inform a Levenberg Marquadt numerical optimizer in Matlab to generate the system model. For a given trajectory, the standard deviation of the

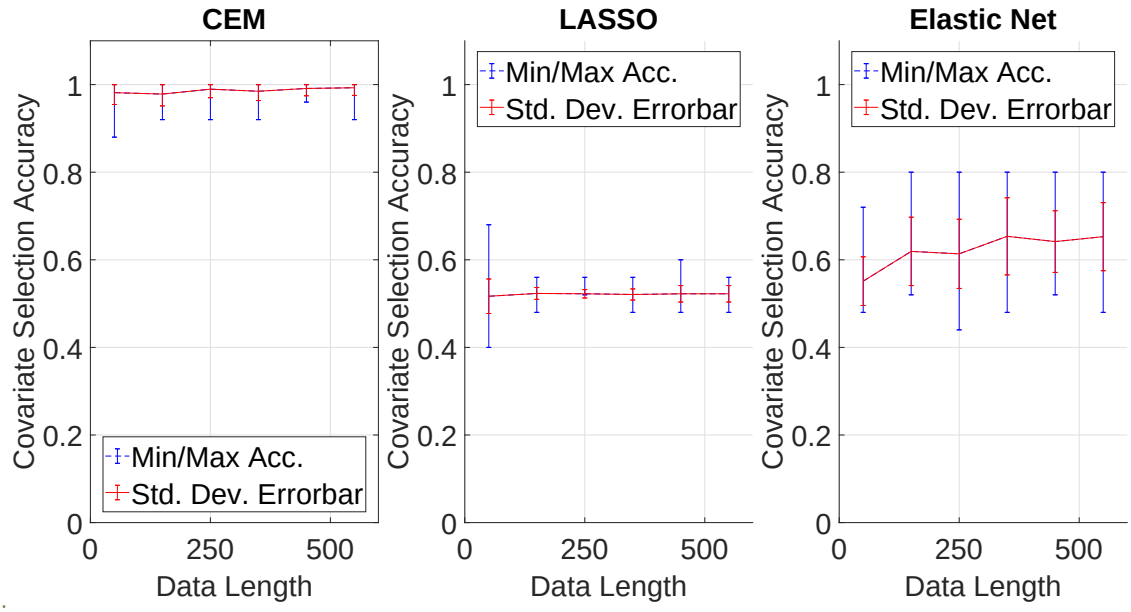

Figure 4.7: CEM, LASSO, and Elastic Net Covariate Selection Performance for Linear System (Red Errorbars: $\pm$ 1 standard deviation, Blue Errorbars: min/max accuracy.)

measurement noise is taken to be the reciprocal of the average state value over the time series multiplied by a scalar noise multiplier n. This scaling of the noise by the average state value serves to keep the signal-to-noise ratio relatively constant among all trials. A total of 40 simulations are generated for processing with LASSO and elastic net, while 15 simulations are processed with the CEM method (due to the longer computation time required). Numerical estimates for the A matrix in Equation (4.14) are obtained directly with LASSO and elastic net for each trial, while in the CEM case the CEM is first used to obtain a reduced set of model parameters and Levenberg-Marquardt optimization [71] is then used to estimate the parameter values.

The average covariate selection accuracy for these Monte Carlo simulations is shown in Fig. 4.9 for varying levels of noise (measured by the multiplier n), as well as for varying training data lengths. As in Figs. 4.7 and 4.8, the CEM exhibits the highest accuracy in the case of no noise. As the noise level increases, all three methods lose accuracy and exhibit comparable performance.

Interestingly, however, the methods by which covariate selection accuracy degrades is

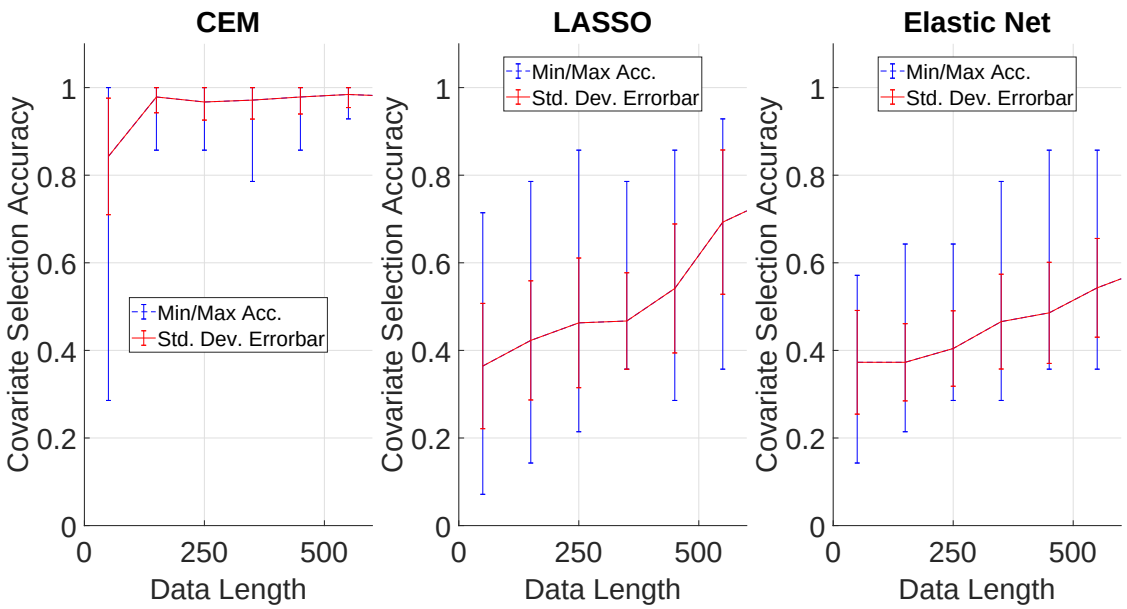

Figure 4.8: CEM, LASSO, and Elastic Net Covariate Selection performance for expanded Van der Pol Oscillator (Red Errorbars: $\pm$ 1 standard deviation, Blue Errorbars: min/max accuracy)

different between the CEM-based technique on one hand, and LASSO and elastic net on the other. This is illustrated in Fig. 4.10, which shows the number of zero entries in the A matrix estimated by each of the algorithms normalized by the total number of entries. The actual system given in (4.14) has 12 zeros out of 25 entries, and thus 48% of entries should be zero if covariate selection is performed perfectly. Figure 4.10 shows that as the noise level increases, the CEM method tends to overpredict the number of zero parameters (i.e., removing too many covariates from the model). As the noise level becomes extremely high, the CEM-based algorithm will tend to eliminate all parameters as noise hides any information transfer between states. In contrast, LASSO and elastic net tend to shrink the model less as noise increases, leading to models with more covariates than necessary.

This difference in how noise affects shrinkage performance has a direct effect on how error in manifested in the resulting model. Generally, as noise increases, models generated from CEM-based shrinkage estimates will be overly sparse, while LASSO and elastic net models will be overly populated. As a result, CEM-derived models will avoid overfitting; however, they will not contain sufficient terms to be able to fully model the dynamics, and

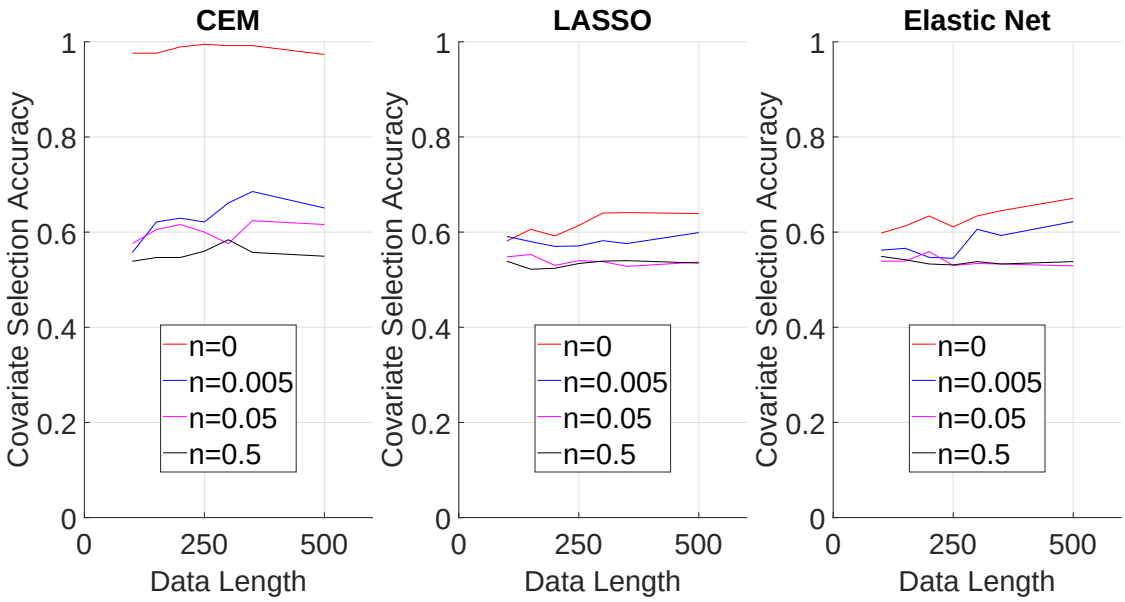

Figure 4.9: Covariate selection accuracy in the presence of measurement noise

thus their predictions can become inaccurate after a short period of time. On the other hand, LASSO and elastic net models have the potential to model the underlying dynamics fairly well with more than the required number of terms included, but this means that the model may be overfit and may poorly predict performance from initial conditions that are significantly different from those used to build the model.

To illustrate this, the models identified using Levenberg-Marquardt (resulting from the CEM selection algorithm) and from LASSO and elastic net for the full Van der Pol system are evaluated in terms of prediction accuracy. For each model identified, random initial conditions are generated using a uniform distribution over $[-3.5, 3.5]$ for each state (similar to the Gaussian distribution used to generate the training data for the shrinkage estimates). The model is then used to generate a state time history of 250 data points. This time history is compared with the ”true” state history from the same initial conditions using the actual model in Equation (4.17) by computing the mean square error. This is repeated for 50 trials, for each of the 15 models generated by the CEM-based algorithm (or 40 models, in the case of LASSO and elastic net). Furthermore, these propagation experiments are performed in two ways. In the first, the actual (true) data point is used to re-center the predicted trajectory

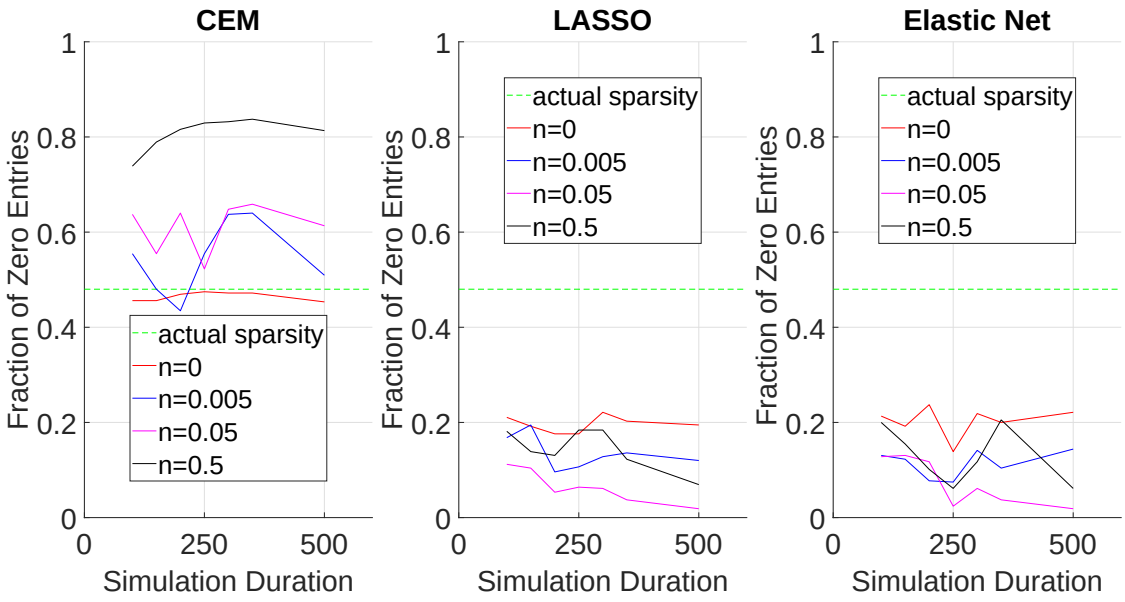

Figure 4.10: Fraction of zero entries in the presence of measurement noise

at each timestep, as if an actual measurement is available. This is similar to how a system model may be used within a Kalman filter. In the second case, the model is used to predict the time history over the entire 250-point sequence without re-centering. These predictions are referred to as ”short-term” and ”long-term” predictions, respectively.

Figures 4.11 and 4.12 show the results of these studies, where the x-axis in each figure refers to the length of training data used to build the model. In the case of no noise, the model predictions resulting from the CEM shrinkage estimates are significantly more accurate for both short-term and long-term predictions, as would be expected given the CEM's superior performance in identifying the true model covariates. However, in the cases of noise for both short-term and long-term predictions, the methods yield similar levels of accuracy, with the CEM-based models exhibiting slightly worse performance. Given the fact that the initial condition distributions for the predictions were similar to those used to build the models (i.e., similar distributions for training and validation), the tendency to overfit will be low and thus this weakness of LASSO and elastic net is not reflected.

A second study is performed identical to the one above except using initial condi-

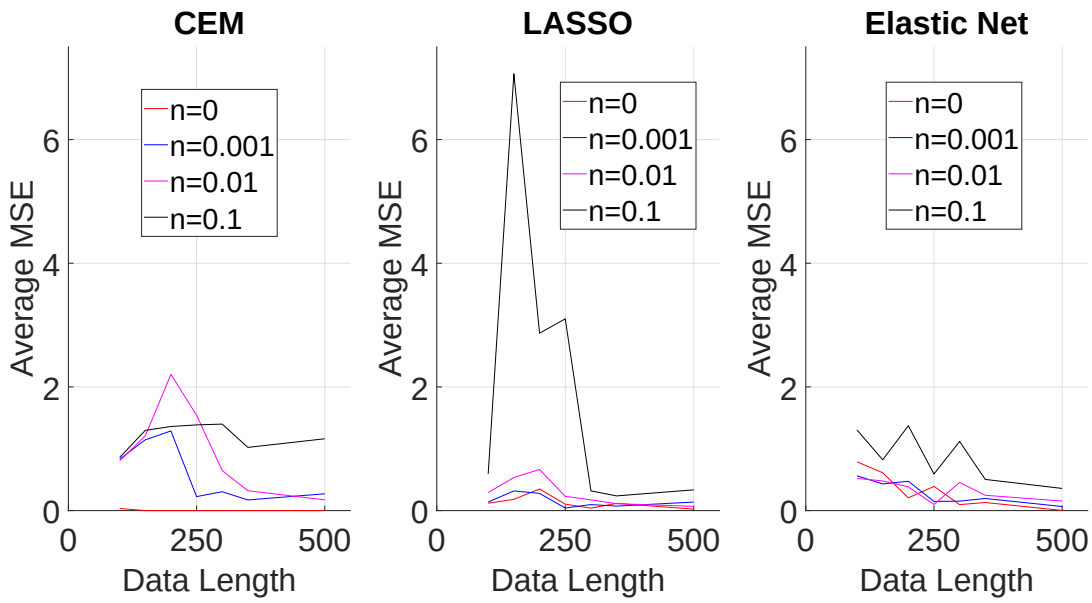

Figure 4.11: Mean square error for short-term prediction using initial condition distributions similar to training

tions for the validation trajectories drawn from a different uniform distribution given by $[-6, -3.5] \bigcup [3.5, 6]$. Here, the validation trajectories start from very different initial conditions than those used in training to develop the models. The results of this study are shown in Figures 4.13 and 4.14. First, note that the average MSE values shown in Figs. 4.13 and 4.14 are significantly higher than that shown in Figs. 4.11 and 4.12 due to the difference in initial conditions used between training and validation data, thus predictive performance is worse. Second, the MSE for the CEM case in Fig. 4.13 is much less than the MSE for LASSO and elastic net when the training data length is short (less than approximately 250 samples), with the differences particularly evident in the higher noise cases. In these situations, the LASSO and elastic net models are overfit with insufficient training data provided to generate an accurate model. Interestingly, in the case of long-term prediction, the results in Fig. 4.14 show that the LASSO and elastic net models are more accurate than the CEM-derived models when noise is present in the data, regardless of training data length. This is because the CEM-derived models do not contain enough covariates to accurately simulate the dynamics of the system, leading to large biases in the trajectory predictions. Overall,

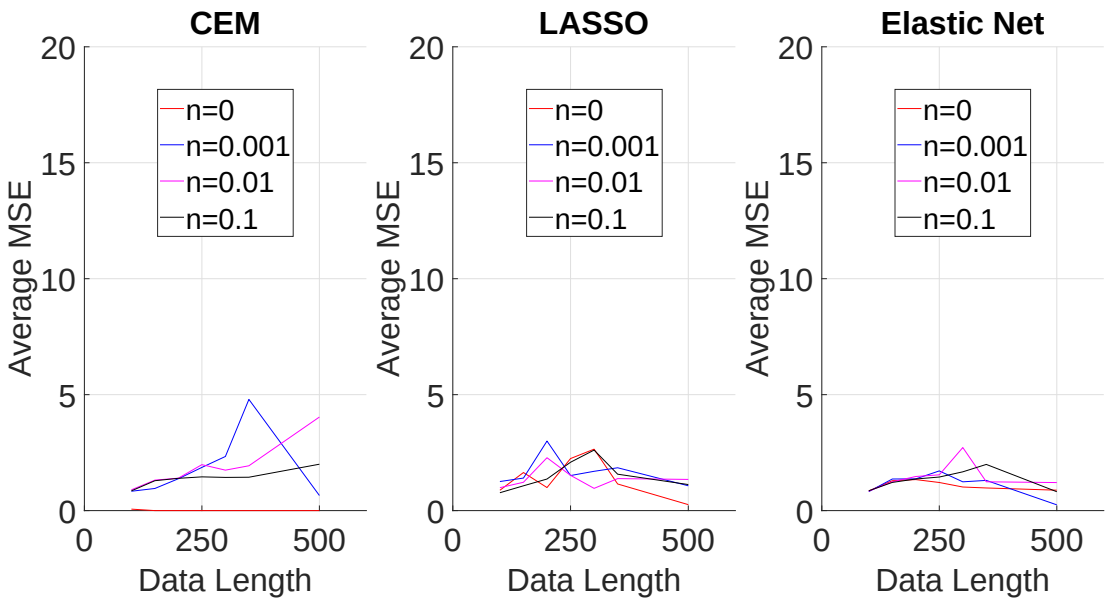

Figure 4.12: Mean square error for long-term prediction using initial condition distributions similar to training

these results show that the CEM-derived models offer better performance for short-term prediction (where biases can be regularly corrected) as they suffer less from overfitting. This is because they tend to be lower order, as more covariates are removed from the model as noise increases. On the contrary, the LASSO and elastic net models provide better long-term prediction (where biases are not corrected) because the retention of more terms in the model allows the dynamics to be modeled more accurately. This outweighs the inaccuracies caused by overfitting.

This section compares the CEM covariate selection algorithm with the commonly-used LASSO and elastic net techniques. Results demonstrated that in the absence of noise, the CEM-based algorithm better estimates the model's structure and yields a lower mean squared error when used for prediction. In the presence of noise, performance of the CEM-based algorithm, LASSO, and elastic net degrade in separate ways. The CEM-based technique tends to remove too many model variables, a result that avoids overfitting but leads to poor modeling of the long-term dynamics. LASSO and elastic net, however, tend to retain too many variables in the model. This leads to potentially better performance in long-term

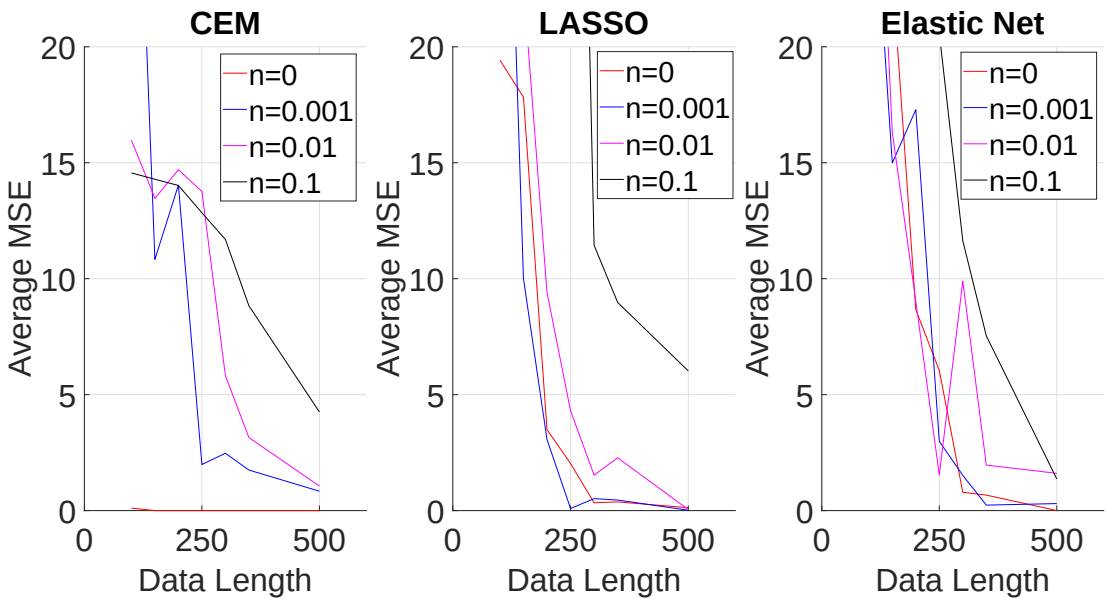

Figure 4.13: Mean square error for short-term prediction using initial condition distributions different from training

forecasting, but also susceptibility to overfitting. Thus, the CEM-based algorithm will be more accurate for short-term prediction scenarios and/or when noise in data is low, but LASSO and elastic net will provide better performance for long-term predictions and/or when the likelihood for overfitting is low due to a highly rich training data set.

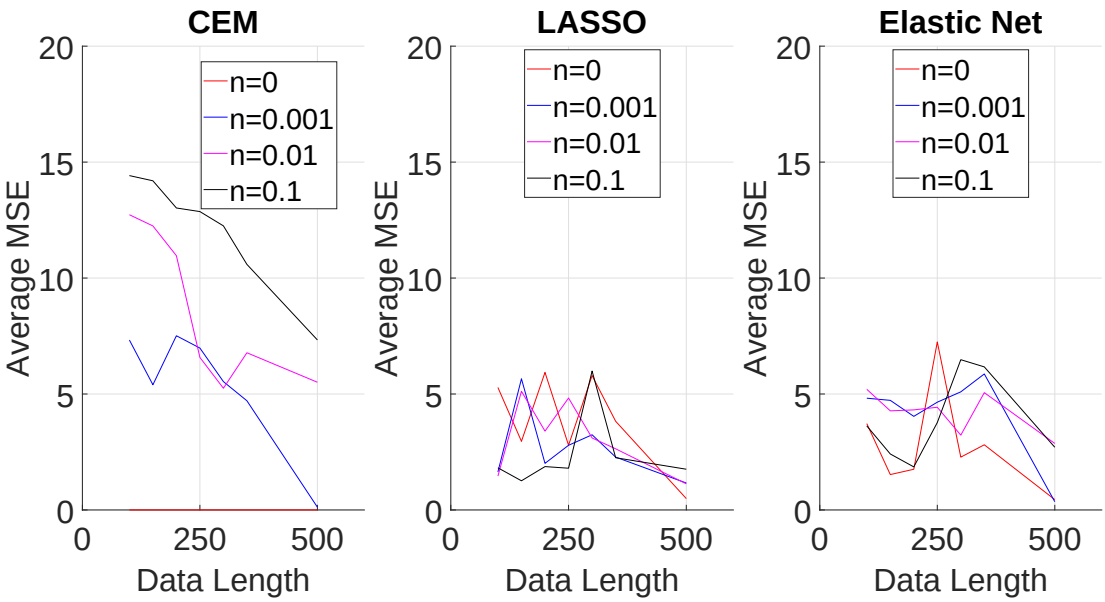

Figure 4.14: Mean square error for long-term prediction using initial condition distributions different from training

CHAPTER 5
PRACTICAL CONSIDERATIONS FOR USAGE OF CAUSATION ENTROPY MATRIX

This Chapter's focus is on providing insight into how to interpret the computed CEM when using imperfect data likely to be encountered during real world applications. First, the effects of noise on the estimation of causation entropy is discussed. This section considers both measurement noise as well as model mismatch, which can be generated by discretization error or unmodeled dynamics in the system. The second half of this section explores the impact of KDE on the estimation of the causation entropy. This includes a study on the bandwidth selection problem as well as how to select the optimal amount of data to include in causation entropy estimation to have a well formed PDF of the underlying dynamics available.

5.1 Noise Considerations when Computing CEM

This section explores the effects of noise and model mismatch on the effects of CEM covariate selection. For the purposes of this work, measurement noise (rather than process noise) is considered, meaning that the system dynamics are assumed to be deterministic; however, real-world sensors introduce noise at each collected data point. In this work, sensor noise is assumed to be drawn from a zero-mean Gaussian distribution. Model mismatch in the context of this work may stem from two major factors – an absence of the proper covariates in the model to adequately describe the system dynamics, and/or error caused by time discretization. The effects of noise and model mismatch will first be explored analytically, and then simulation results will be presented to illustrate the trends uncovered through analysis. First, an overview on the general effects of measurement noise and discretization error are introduced with insight into the manner of CEM degradation provided.

Then, a detailed proof of the reason for CEM performance degradation in the presence of measurement noise is provided along with a study on the pattern of CEM degradation in the presence of measurement noise. Subsequently, details on the effects of discretization error and unmodeled dynamics are shown with a trade study run on the case of discretization error and a mathematical decomposition of the nature of CEM degradation in the case of unmodeled dynamics.

5.1.1 Measurement Noise and Sampling Rate Based Error: An Overview

This section considers the effect of noise and unmodeled dynamics on a Van Der Pol oscillator system with continuous time equations given by [70],

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= 1.15x_2 - x_1 - \mu x_2 x_1^2 \end{aligned} \tag{5.1}$$

In order to test the performance of the model shrinkage techniques, a modified Van Der Pol oscillator model is used in which the candidate state function vector $\mathbf{F}$ is augmented with extraneous functions x_1x_2, $x_1x_2^2$, x_1^2 and x_2^2. The oscillator equations are transformed to discrete-time using the transformation technique described in [56] with a time step of $T = 0.01$ sec, yielding a model of the form shown in (1.11). When discretized and written in the form of Eq. (1.11), the standard Van Der Pol model has 6 total parameters of which 5 are nonzero and only one is zero. This discrete model is shown in Eq. (5.2), where $[\cdot]_t$ denotes that the vector elements are evaluated at time step t. The modified model has 14 total parameters when discretized, of which the same corresponding 5 parameters are nonzero and the remaining 9 parameters are equal to zero. This modified model is shown in discrete form in Eq. (5.3). In the results shown below, the model in Eq. (5.1) is used to generate data, and the goal of the CEM is to identify the 9 zero parameters in Eq. (5.3) for removal prior to parameter optimization.

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{t+1} = \begin{bmatrix} 1 & T & 0 \\ -T & (1.15T+1) & -\mu T \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_2 x_1^2 \end{bmatrix}_t \tag{5.2}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{t+1} = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 & 0 \\ -T & (1.15T+1) & -muT & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_2 x_1^2 \\ x_1 x_2 \\ x_1 x_2^2 \\ x_1^2 \\ x_2^2 \end{bmatrix}_t \tag{5.3}$$

The results in this section are generated by simulating data from Eq. (5.1) using ode45 and sampling the results using different timesteps T, and by adding measurement noise to the resulting state time histories. Figure 5.1 shows an example trajectory from nonzero initial conditions using $T = 0.01$. Specifically, Fig. 5.1 (top) shows typical perturbations to the state measurements caused by zero-mean Gaussian measurement noise with a noise multiplier of 0.5, while Fig. 5.1 (bottom) shows a comparison between the state time histories of the model in (5.3) using $T = 0.01$ with a fourth order Runge Kutta solver and a when using a fairly large time step of $T = 0.1$ sec when forward propagating the proposed discretized model. In both plots of Figure 5.1, the solid lines represent the "true" state dynamics, i.e., the result of integrating the continuous-time equations of motion. However, in the plot on the bottom the dashed lines represent the discrete model that the CEM is attempting to fit. As the time step is increased, the finite difference approximation increasingly deviates from the true derivative, and thus the discretized model and the continuous time solution will diverge. In both the case of measurement noise and model mismatch, the data collected through sampled measurements deviates from the idealized model and leads

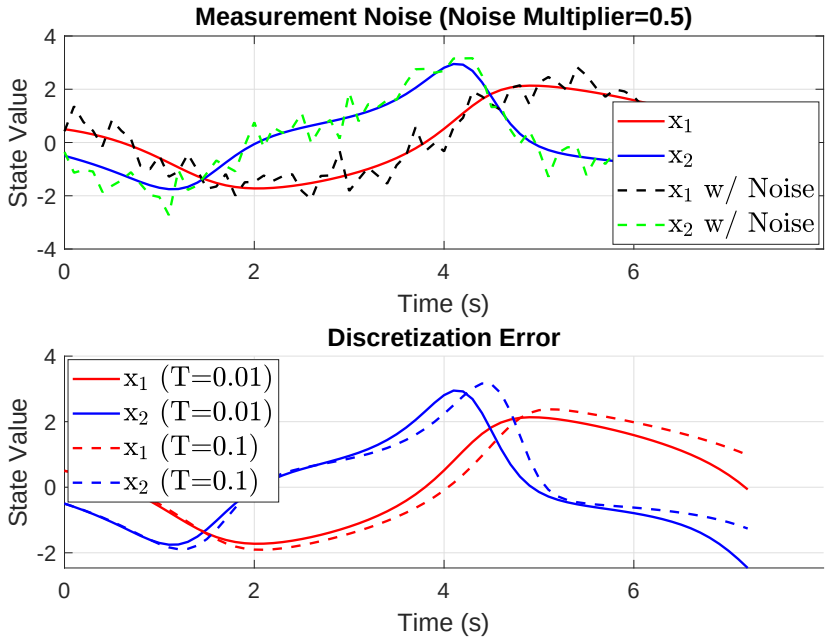

Figure 5.1: Trajectory Disturbances Caused by Noise and Time Discretization.

to a loss of accuracy in the CEM. It is worth noting that measurement noise and model mismatch cause decay in the accuracy of the CEM in different ways.

In order to quantify the performance of the CEM, an error metric is introduced. The metric is defined below in Equations (5.4-5.8). α is the average Causation Entropy of the entries that should be nonzero. $\beta(i,j)$ is defined as representing correct nonzero entries as in Equation (5.4), $\psi(i,j)$ is the number of correct zero entries in the CEM as in Equation (5.5), $\phi(i,j)$ is false negatives as defined in Equation (5.6), and $\upsilon(i,j)$ represents false positives as in Equation (5.7). As in Equation (1.11), $\Theta(i,j)$ refers to the (i,j) entry of the actual parameter matrix.

$$\beta(i,j) \begin{cases} 0 & \Theta(i,j) = 0 \\ 1 & \Theta(i,j) \neq 0 \ \& \ CEM(i,j) \neq 0 \end{cases} \tag{5.4}$$

$$\psi(i,j) \begin{cases} 0 & \Theta(i,j) \neq 0 \\ 1 & \Theta(i,j) = 0 \ \& \ CEM(i,j) = 0 \end{cases} \tag{5.5}$$

$$\phi(i,j) \begin{cases} 1 & \Theta(i,j) \neq 0 \ \& \ CEM(i,j) = 0 \\ 0 & otherwise \end{cases} \tag{5.6}$$

$$\upsilon(i,j) \begin{cases} 1 & \Theta(i,j) = 0 \ \& \ CEM(i,j) \neq 0 \\ 0 & otherwise \end{cases} \tag{5.7}$$

Using these definitions, the error metric ($E.M.$) is given in Equation (5.8)

$$E.M. = \sum_{i=1}^{m} \sum_{j=1}^{n} CEM(i,j)\beta(i,j) + \alpha\psi(i,j) - \alpha\phi(i,j) - \frac{1}{\alpha} CEM(i,j)\upsilon(i,j) \tag{5.8}$$

This error metric has several meaningful characteristics. First, it provides increased rewards for correct entries when the entry has a large magnitude, suggesting certainty that the parameter should be nonzero. Second, it provides a higher reward if zeros appear in correct locations in the CEM when the correct nonzero entries in the CEM have a larger magnitude, which suggests that the estimator is both accurate and discriminatory. Third, the penalty increases for false negatives when correct, nonzero parameters in the CEM have large magnitudes as the large magnitudes suggest incorrect certainty of the estimator. Finally the metric provides additional penalty in the case when there is a small difference between false positives magnitudes and correct parameter magnitudes as this would make it difficult for the user to make informed decisions based on entries in the CEM. When considering this error metric, a higher metric value corresponds to better performance by the CEM. The average error metric value on the computed CEMs from the Van Der Pol oscillator experiment can be visualized as in Figure 5.2. When the measurement noise magnitude and time step used are small, the error metric returns a very large value, implying strong CEM performance. As the measurement noise magnitude and time step duration increase, the metric approaches zero. Thus, the CEM clearly has a far better performance in cases where there is little model mismatch and low amounts of measurement noise. However,

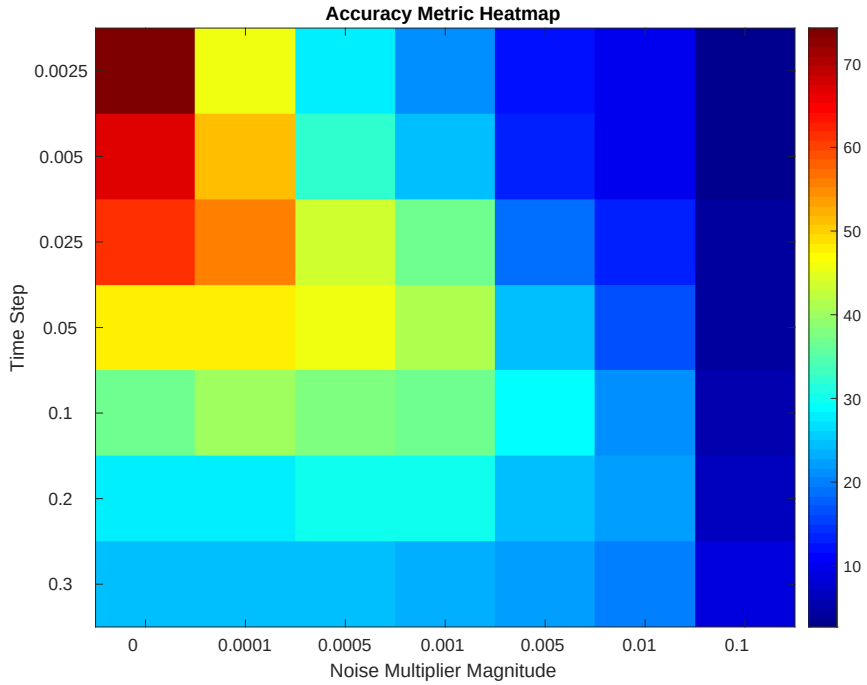

Figure 5.2: Error metric values for various time steps and noise multipliers

the mechanisms by which measurement noise and discretization error cause the CEM to degrade are discussed below.

In order to study the consequences of measurement noise and discretization error, the effects of sampling rate and noise are studied simultaneously. Ten random sets of initial conditions were generated and the trajectories computed using Eq. (5.3) for a particular noise multiplier and time step T. The CEM was then computed from each trajectory and compared to the parameter matrix in (5.3) by computing the accuracy metric, number of false negatives, and number of false positives. These metrics were averaged together over the 10 runs and plotted for each combination of noise multiplier and time step in Figs. 5.2, 5.3, and 5.16.

5.1.2 Measurement Noise

When measurement noise is added to the state values, the resulting time series is fractionally comprised of ”signal” and noise portions. As measurement noise increases with

respect to the state values being measured, the noise content can start to dominate the time series values, reducing the signal-to-noise ratio close to zero. To study this analytically, consider a three-state system in the limiting case when the state time histories are completely comprised of noise. Specifically, let the three states X, Y and Z be independent, Gaussian random processes. The causation entropy from X to Y given Z can be written as conditional entropies according to [18],

$$C_{X\to Y|Z} = H(X|Y) - H(X|Y,Z) \tag{5.9}$$

However, because of the independence of X, Y, and Z, $H(X|Y) = H(X)$ and $H(X|Y,Z) = H(X)$ and thus $C_{X\to Y|Z} = 0$.

Thus, as the noise level perturbing state measurements increases, the CEM values will generally become smaller since the difference between the conditional entropies in (5.9) will shrink. As the noise becomes large and eventually dominates the signal, the CEM values will all approach zero, regardless of the underlying causality in the system dynamics. This leads to a tendency for the CEM to produce so-called "false negatives" in the presence of significant measurement noise, i.e., values of the CEM that are estimated as zero even though the associated parameter in the system model is nonzero. If the CEM is used to reduce the parameter set as discussed in [56], this can lead to the removal of too many parameters and a model that cannot adequately capture the system dynamics. Fortunately, as will be shown later in this Chapter, standard filtering techniques can be used to reduce noise and improve the signal-to-noise ratio in the timeseries, leading to improved CEM estimates.

The tendency of the CEM towards 0 in the presence of measurement noise is clearly visible in Figure 5.3. The figure demonstrates the appearance of false negatives in the study used to generate the error metric study in Figure 5.2. When the driving source of error is the large amount of noise (rather than discretization), false negatives become increasingly prevalent as the entire CEM approaches zero. As will be demonstrated later in Figure 5.16,

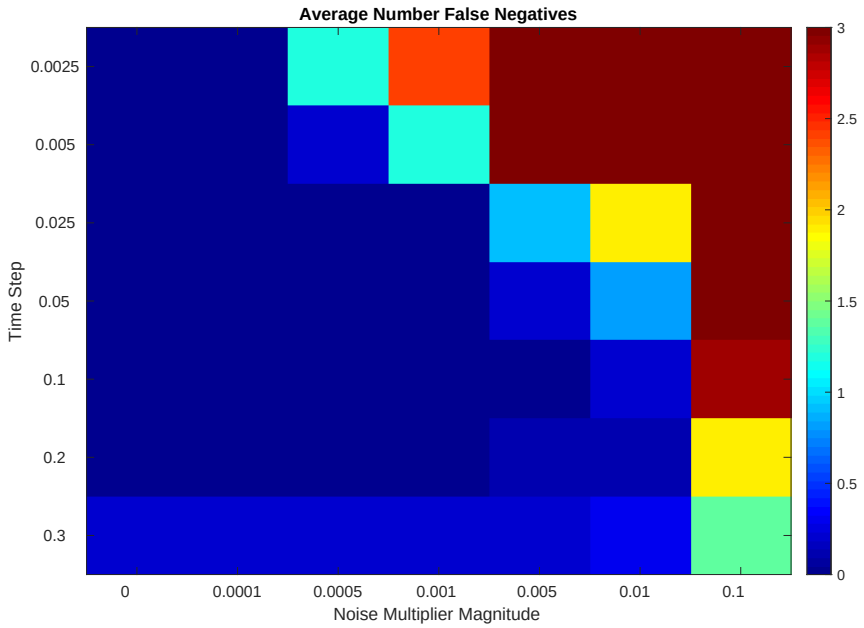

Figure 5.3: Average number of false negatives

in the presence of large magnitude measurement noise there is almost zero appearance of false positives as predicted.

Causation Entropy Magnitudes and Sensitivities of Lost Parameters in the Presence of Measurement Noise

The previous section detailed the tendency of the CEM to report false negatives in the presence of measurement noise. This section seeks to explore the behavior of the CEM as it tends towards the zero matrix to identify if there is a discernible pattern in how the parameters lost disappear. This section seeks to demonstrate that the parameters are lost in order of least important first to most important last where importance is defined by the causation entropy magnitude, which is proportional to the parameter sensitivity. In order to study this, the inverted pendulum from Section 3.1.2 and the harmonic oscillator used in Section 3.2.2 are considered.

To explore this, denote a parameter that is associated with a false negative as a "lost parameter", since the CEM was no longer able to identify its importance to the model at a

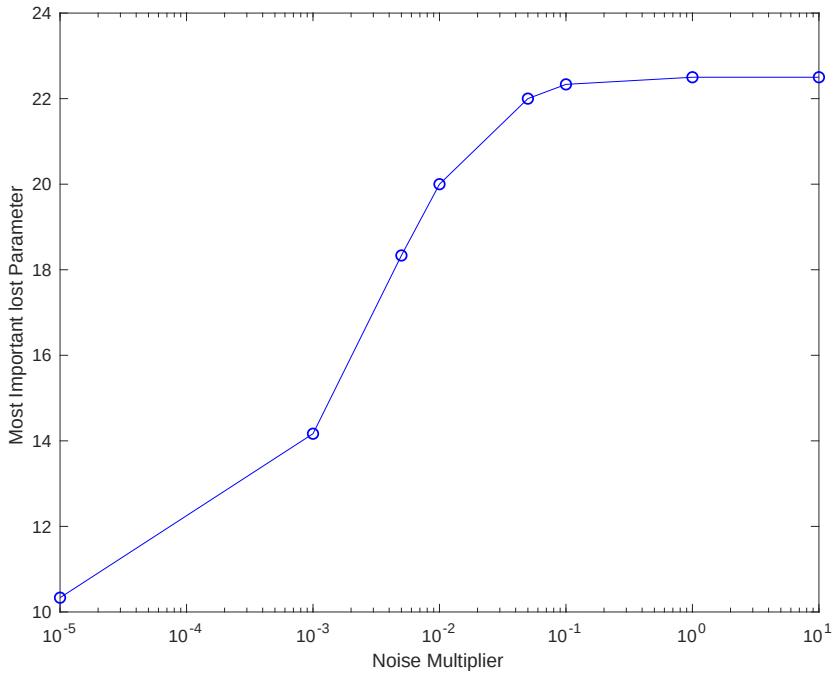

Figure 5.4: Mass-Spring-Damper average importance of top six most sensitive parameters lost

given level of noise. Trajectories were simulated with varying levels of non-dimensionalized noise added as discussed in Section 5.1.1. The parameters in the no noise case were then ranked according to their sensitivity such that the higher the ranking the higher the sensitivity. The sensitivity ranking of each lost parameter was recorded per given level of noise. For the mass-spring-damper, the ranking of the 6 highest-sensitivity lost parameters was averaged. For the inverted pendulum, the ranking of the 3 highest-sensitivity lost parameters was averaged (less parameters were averaged since the pendulum has fewer overall model parameters). Note that if, at a given noise level, there were less than this number of lost parameters to average, the average over all lost parameters was used. The results of this analysis are shown in Figures 5.4 and 5.5. These figures demonstrate that as the noise level increases, the average sensitivity (the importance) of the parameters lost increases. The parameters in the CEM disappear as the noise level increases in order of least important first, at low noise levels, to most important, at very high noise levels.

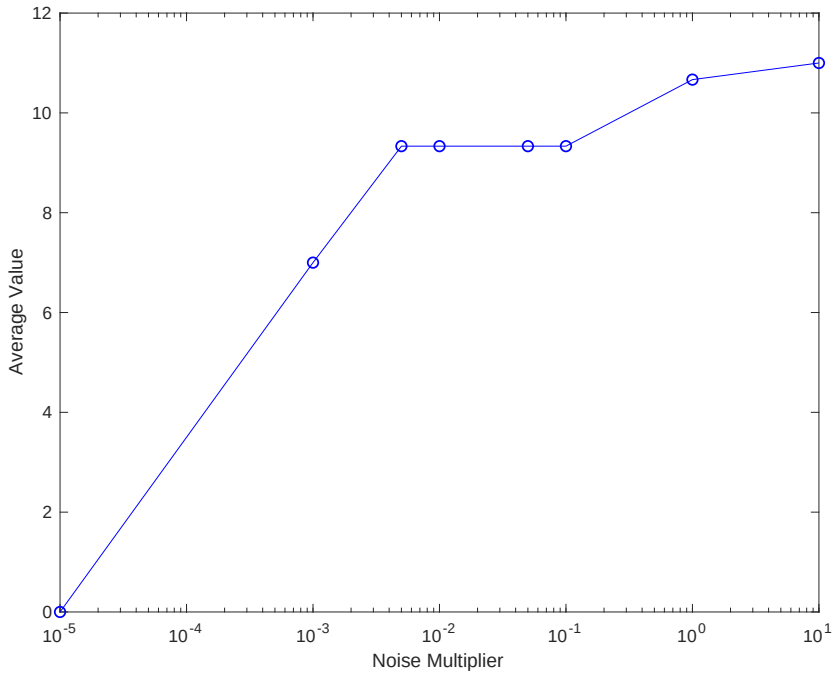

Figure 5.5: Inverted Pendulum average importance of top three most sensitive parameters lost

A parallel analysis is performed to understand the significance of the element magnitudes in the CEM. Figures 5.6 and 5.7 represent the same analysis as in Figs. 5.4 and 5.5, except that the ranking is done in terms of the magnitude of the causation entropy values in the CEM as opposed to the sensitivity value of the corresponding model parameter. Comparing Figs. 5.6 and 5.7 and 5.4 and 5.5, the sets of figures look nearly identical. This observation solidifies the conclusion of Section 3.2 that the magnitude of the causation entropy values in the CEM is directly connected to, and can be used as a proxy for, the sensitivity value associated with the corresponding model parameter. Additionally, parameters lost first to small noise will be parameters with low causation entropies and thus low sensitivities.

This outcome is encouraging as it suggests in the presence of noise that the first parameters to be lost will be those that are least important, so hopefully the achieved model will still be relatively accurate as the truly dominant terms will still be included.

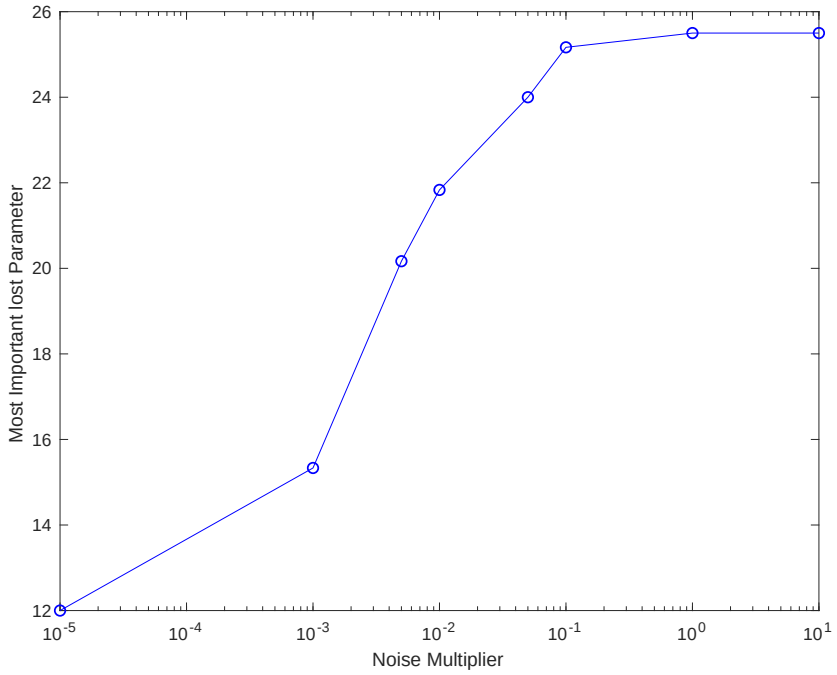

Figure 5.6: Mass-Spring-Damper average importance of top six highest CE parameters lost

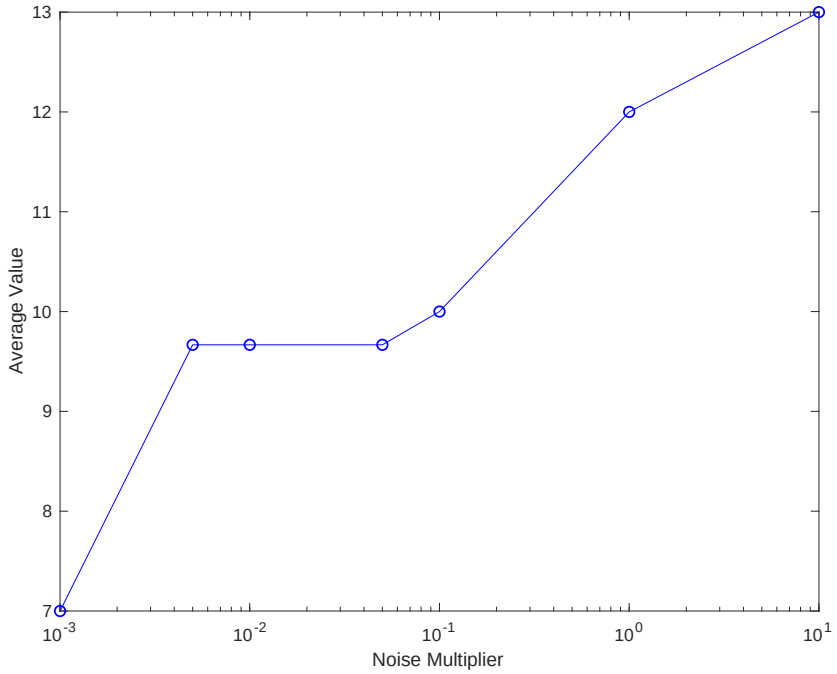

Figure 5.7: Inverted Pendulum average importance of top three highest CE parameters lost

5.1.3 Data Smoothing in Presence of Measurement Noise

Previous sections have discussed the fact that as measurement noise of increasing magnitude is added to a system, the CEM will tend towards a zero matrix as the noise obscures the information flow and makes the random variables appear independent of one another regardless of any relationships that exists.

The destructive nature of noise is not unique to the proposed CEM methodology and is highly problematic for many identification and optimization problems. In many different engineering applications, filtering or smoothing is used to remove random noise from signals, which is nearly identical to the case of measurement noise in the case of mechanical systems. The moving average filter is a common low pass filter used in the time domain due to its easy implementation and success at removing random noise [72]. Moving average filters can be designed as either causal or non causal depending on the set of data points used. This work will focus on non-causal filters (often also referred to as smoothers) as they tend to give better results and do not induce a phase lag [73]. As the CEM is not intended for use in real-time applications, access to the entire dataset is available and thus a non causal filter is implementable. In particular, a symmetric, non causal moving average filter given by Equation (5.10) is used. Based on the notation below, $x^{(n)}[j]$ refers to the filtered value for datapoint j using a moving average filter with windowsize n. Note that the floor function $floor(x)$ returns the value of x rounded to the nearest whole number towards $-\infty$.

$$x^{(n)}[j] = \frac{1}{n} \sum_{i=(j-floor(\frac{n}{2}))}^{(j+floor(\frac{n}{2}))} x[i] \tag{5.10}$$

Results of Using a Moving Average Smoother

This section presents the results of using a moving average filter on simulated noisy data before computing the CEM. When using a moving average smoother, the larger the win-

dow, the greater the smoothing achieved with higher frequency noise removed. However, if the window becomes too large, actual information from the signal can also be smoothed out and lead to decreased performance and a large difference between the filtered results and the true nominal trajectory. In order to study this, the extended Van Der Pol oscillator system from Equation (5.3) was used. Zero-mean, Gaussian measurement noise with a scalar noise multiplier of 0.1 was added to the simulated data. Moving average filters were then applied with varying size windows. Note that a moving average filter of window-size one will return the original, noisy data. Figure 5.8 contains the noiseless data and the (correctly) returned CEM structure. Figure 5.9 shows the effects of noise on the data and the

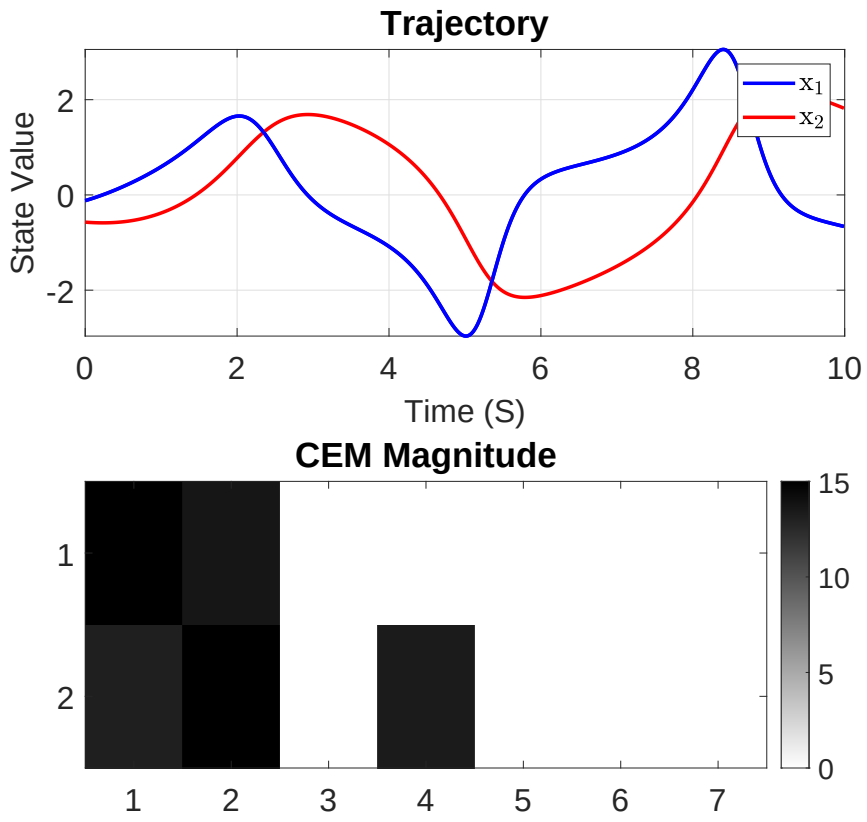

Figure 5.8: True Van Der Pol trajectory with corresponding CEM

CEM. As a window-size of 1 corresponds to the original data, the filtered and noisy data are identical. As expected, the CEM is more sparse than it should be due to the effects

of the measurement noise with only the two largest entries from the noiseless case CEM identified. As the size of the window is increased, the noise begins to be filtered out and a trajectory approaching that of the true trajectory emerges, which causes CEM performance to improve. Figure 5.10 demonstrates the effects of a filter with window-size of

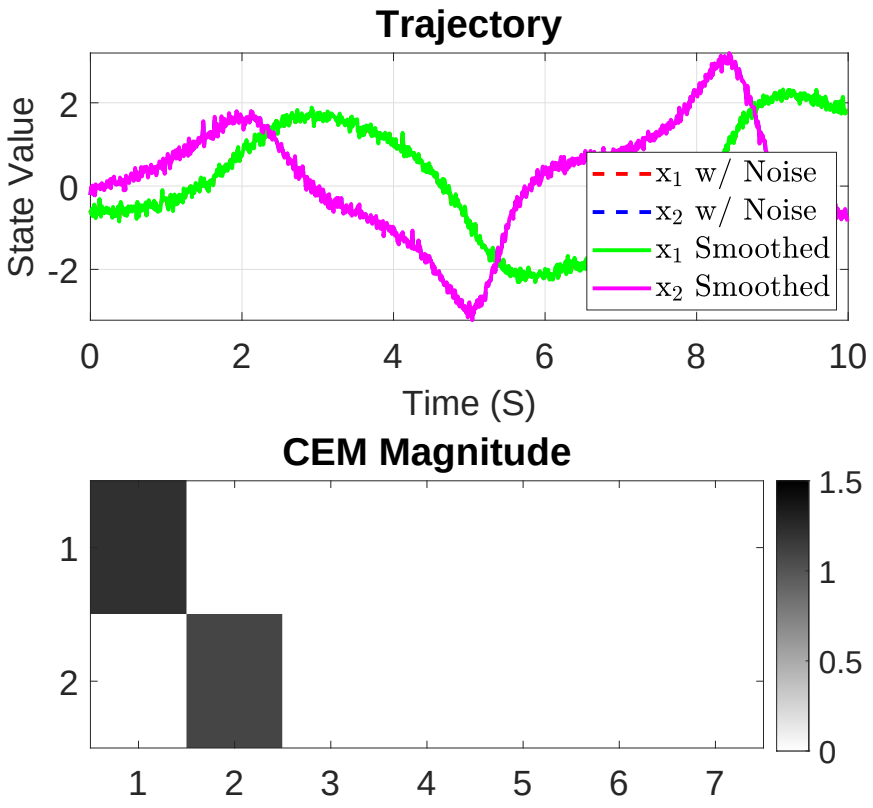

Figure 5.9: Noisy and filtered data with window-size 1 and corresponding CEM

15, which improves the data, but is not sufficient to remove all of the noise. One can see that the filtered data is much smoother than the noisy data and the CEM does recover one previously-missing, correct entry; however, some relationships are still obscured and the CEM is still more sparse than it should be. Increasing the window increases the smoothing characteristics as shown in Figure 5.11. In Figure 5.11, the filtered data is smooth and essentially in the center of the noisy data where the true data from Figure 5.8 is. With this filtered data, the CEM is able to correctly identify the sparsity structure. However, if too

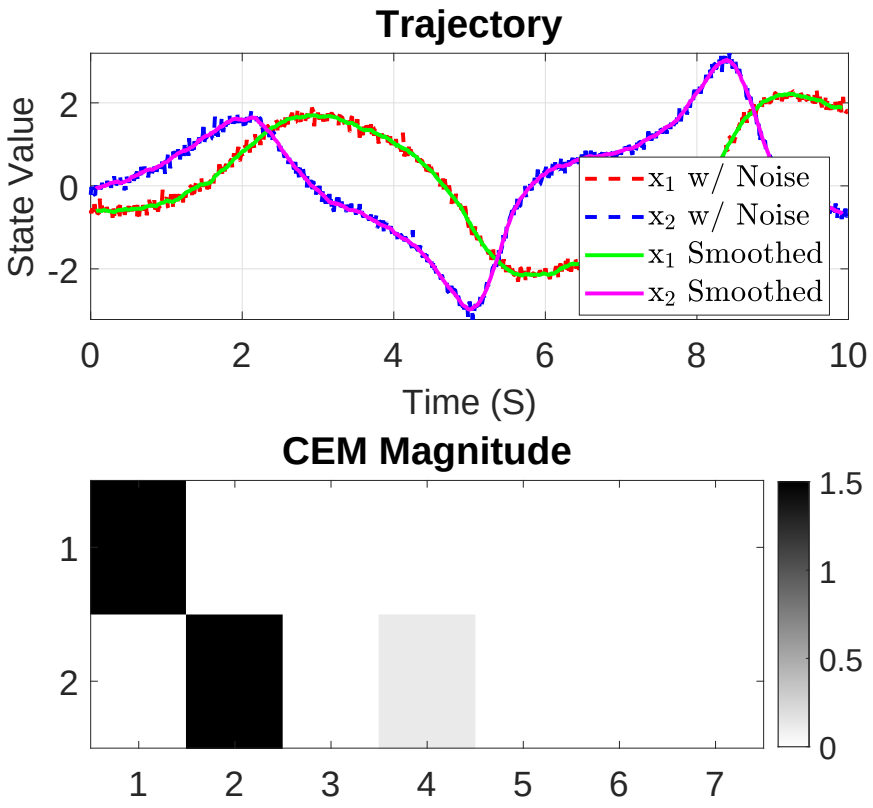

Figure 5.10: Noisy and filtered data with window-size 15 and corresponding CEM

large of a window is used, information from the signal itself can be lost as shown in Figure 5.12.

In Figure 5.12, the filtered data has clearly deviated from the true and noisy data both. The smoothing has not only removed noise, but also altered the actual dynamics of the system as well. This causes a phenomenon nearly identical to that of the case of an improperly large time step as in Section 5.1.4 or as in the case of unmodeled dynamics in Section 5.1.5. This case, as in both of those, represents a situation where there is significant mismatch between the model being fit and the generative dynamics, which can lead to an overly dense CEM. Figure 5.12 has multiple incorrect nonzero entries appearing in the bottom row with all seven potential functions being used when only three are actually necessary.

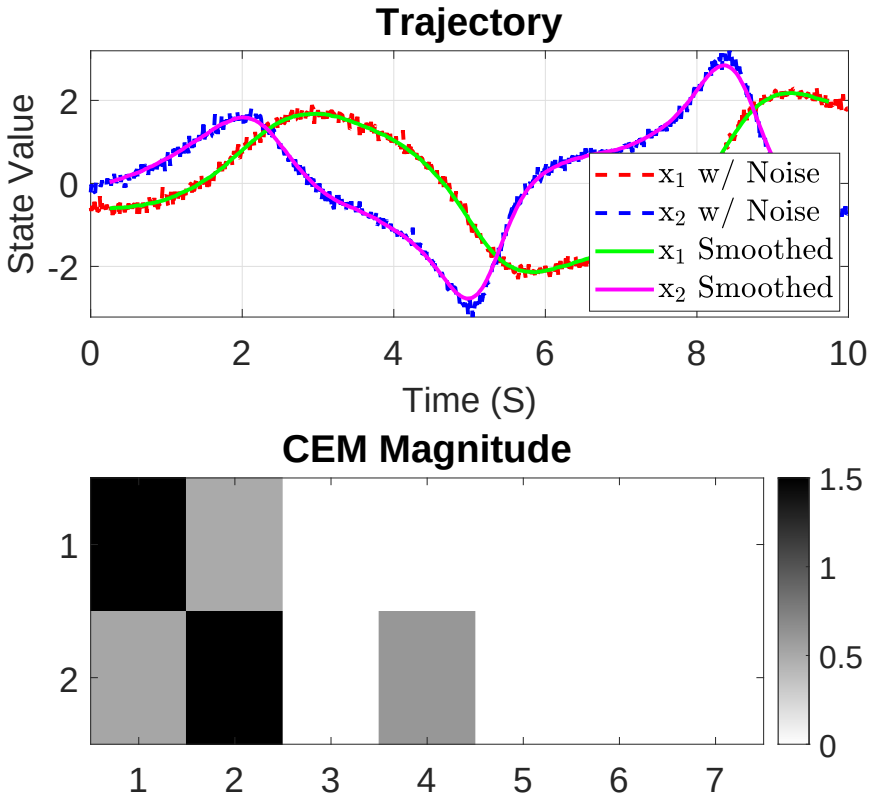

Figure 5.11: Noisy and filtered data with window-size 51 and corresponding CEM

Unintended Considerations from Filtering

The previous section demonstrated the potential to use low pass filtering to mitigate the effects of measurement noise on CEM accuracy. However, there are a couple of considerations when using the CEM post filtering that relate to the amount of data required. The first consideration is a rather straight forward consideration when using filtering in that some portion of the data will be removed as there is not enough data either preceding or succeeding a data point to adequately populate the filter. This is most easily seen in Figure 5.12 as the filtered trajectory curve is missing significant data on each tail as shown by the light blue and green dashed lines starting and ending far before the noisy counterparts. This phenomenon is true in all filtering cases, though less easily visible as the number of data points lost is equal to the size of the filter window with half missing from the beginning

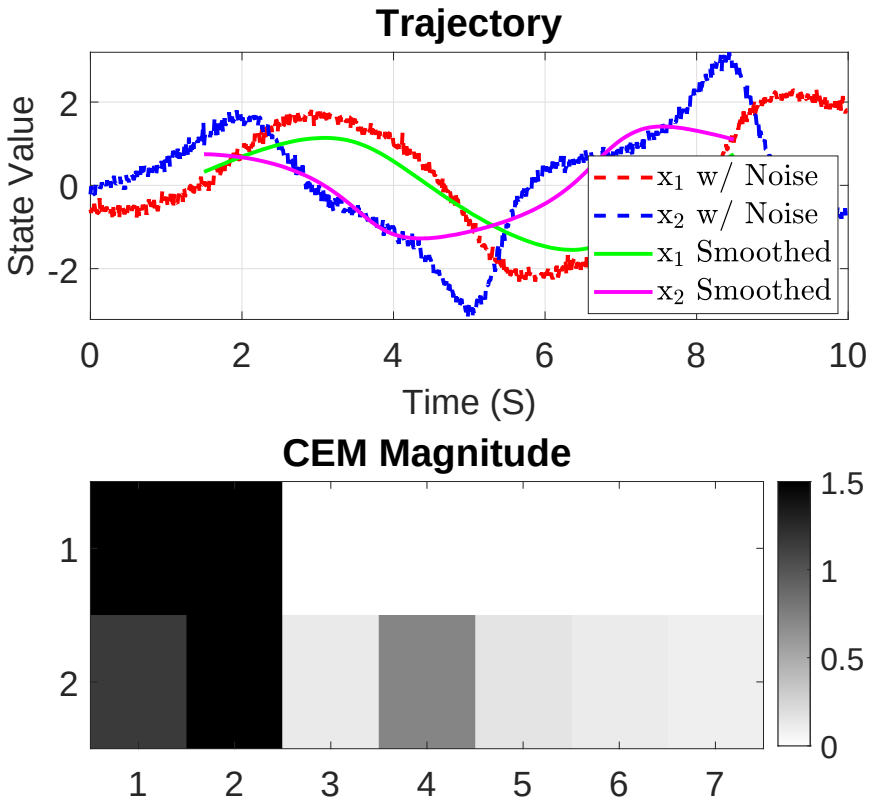

Figure 5.12: Noisy and filtered data with window-size 301 and corresponding CEM

and the other half at the end. Therefore, the user must consider if there is sufficient data to be able to fit the model once the data lost to training the filter has been removed.

A secondary consideration is that the loss in model accuracy due to the deviation from the filtering requires more data than the usual case of no noise in order to be able to correctly identify the sparsity structure of the system. This fact can be seen in Figure 5.13, which shows it takes the filtered data a longer period of time to correctly identify the sparsity structure. In order to generate the plot, the system was simulated with the same noise properties as used in the previous section. However, this time a filter window-size of 101 data points was used to filter the data. The CEM was then computed at various time intervals and the CEM covariate selection accuracy computed for the noiseless data, unfiltered noisy data and filtered noisy data corresponding to the same raw data points. Thus, the

filtered data set will be shorter than the clean and unfiltered data sets.

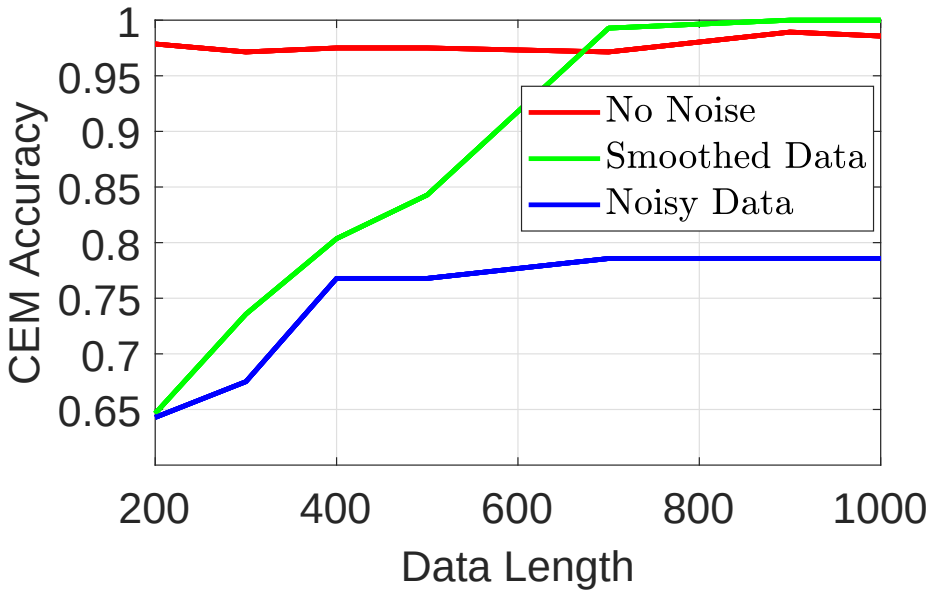

Figure 5.13: Comparison of clean, unsmoothed noisy and smoothed noisy data and corresponding CEM accuracy verses data length

However, one can clearly and encouragingly see that appropriately filtering the data does no worse than that of computing the CEM on the unfiltered data regardless of the number of data points available.

Figure 5.14 demonstrates the effect of the window-size on the CEM accuracy. Figures 5.14 and 5.15 were created by averaging the results over 20 studies using randomly selected initial conditions. The left subplot demonstrates that it takes a window-size of approximately 51 data points to be able to appropriately filter down the noise to achieve high accuracy. The system then has a relatively high accuracy for a large set of window-sizes oscillating between 13 and 14 out of 14 correct identifications of the sparsity structure with the failure mode being an unnecessarily dense CEM as demonstrated by the lower percent sparsity shown on the right hand subplot of Figure 5.14. It is also worth noting, and largely predictable as shown in Figure 5.15 that the filtered and noisy cases have smaller average magnitudes for nonzero entries than that of the noiseless case as both cases have adulterated relationships between the potential model and observed data as there is some slight

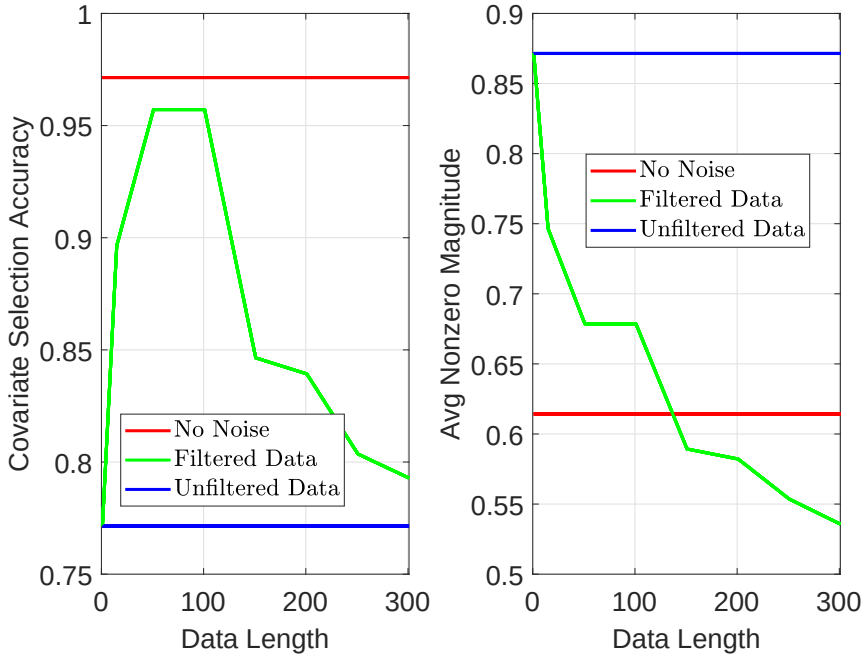

Figure 5.14: Comparison of clean, unfiltered noisy and filtered noisy data and corresponding CEM accuracy verses time for varied filter window-size

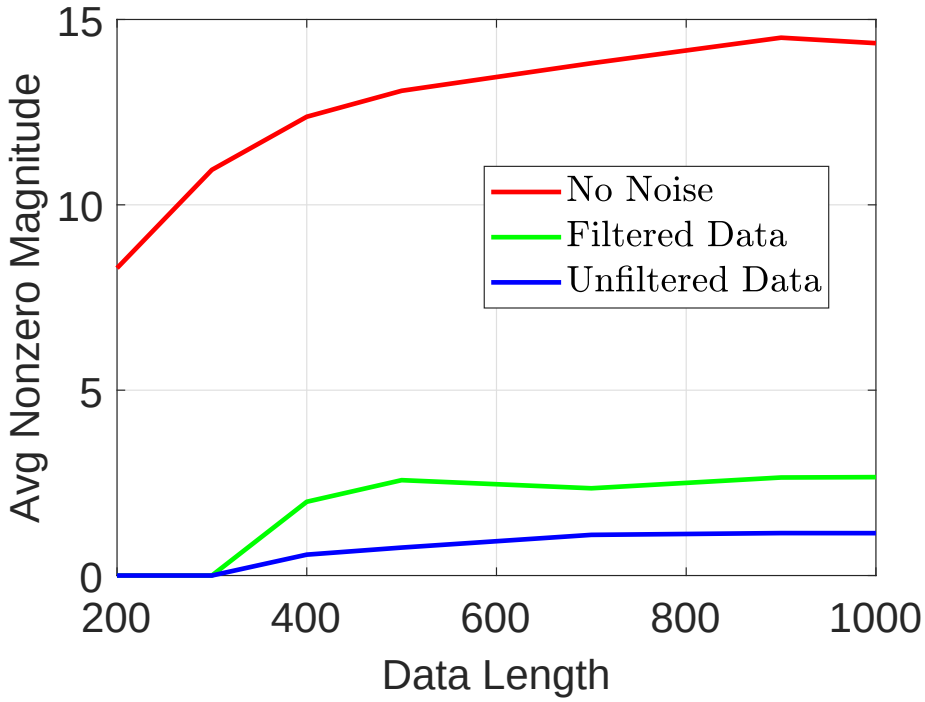

Figure 5.15: Comparison of clean, unfiltered noisy and filtered noisy data average magnitude of nonzero parameters from corresponding CEM verses data length

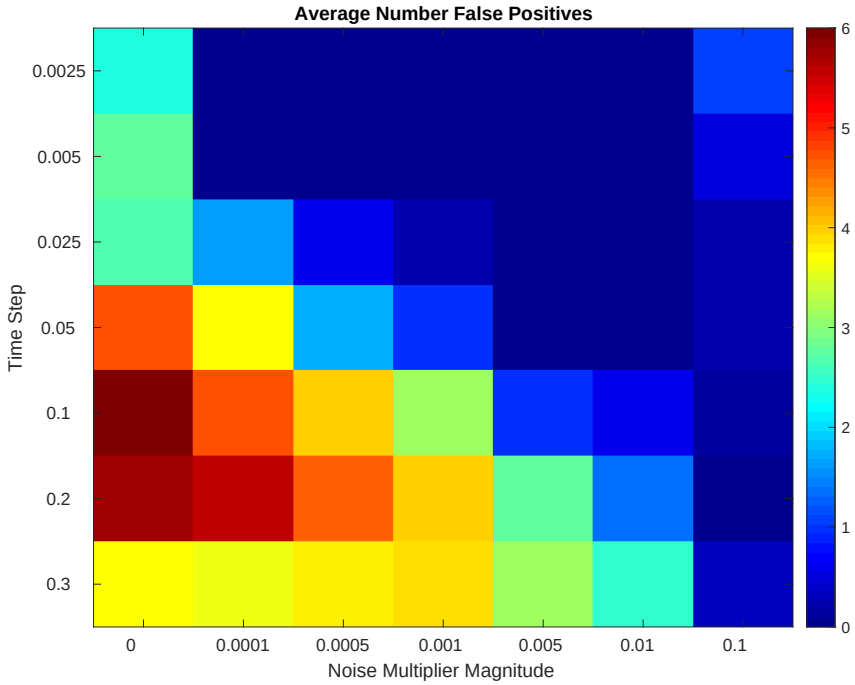

Figure 5.16: Average Number of False Positives for Van Der Pol Oscillator Cases.

deviation between the ideal and actual data due to either the filter, the noise, or both. However, like in the case of the CEM accuracy, the filtered case always had the larger average nonzero magnitude (and higher accuracy), both of which are desirable traits to have.

5.1.4 Model Mismatch

An insufficiently small time step has the effect of creating error in the derivative approximation used in discretization, and thus the discrete-time system will no longer closely match the continuous dynamics, leading to error in the CEM. However, in the case of a large time step, the tendency of the CEM structure is towards false positives. Figure 5.16 shows that, as the time step grows, the number of false positives increases. This is because use of a larger time step creates a mismatch between the model used by the CEM and the actual dynamics. This figure was generated from the error metric study in Figure 5.2 and is identical to Figure 5.3, except the prevalence of false positives is reported instead of that of false negatives.

The results in Fig. 5.16 imply that, as the time step in the model is increased and the discrete model becomes a poorer approximation to the continuous-time dynamics, the extra model terms in Eq. (5.3) that are not present in Eq. (5.2) actually are estimated to provide information to the state updates by the CEM. This means that a model that includes nonzero parameters in Eq. (5.3) corresponding to the false positives in the CEM should be more accurate with large values of T, compared to the actual model in (5.2) with the same value of T. To verify this, two Monte Carlo simulations were performed as follows. First, 100 simulations of the Van Der Pol oscillator in Eq. (5.2) were performed from random initial conditions using a very small time step of 0.01. Then, a Levenberg-Marquardt optimization routine [71] was used to optimize the system parameters for the systems in Eqs. (5.2) (”actual” Van Der Pol model) and (5.3) (”expanded” Van Der Pol model) using a large time step of 0.2 sec. No noise was included in these simulations. The average mean squared error (MSE) over the trajectories using the actual Van Der Pol model was 1.1784, while the average MSE over the trajectories using the expanded Van Der Pol model was 0.1304. The much lower MSE when using the expanded Van Der Pol model shows that the extra terms included in expanded model are actually useful in providing a better approximation to the dynamics with this very large time step. In other words, at such a large time step, the expanded model provides a better approximation of the continuous-time dynamics that the actual discretized model in (5.2). This explains why the CEM produces false positives for certain parameters as the discrete system incurs more modeling error.

One possible explanation for the above results is that the models are overfit – the inclusion of more model parameters in the expanded model allows the optimization process to fit the single trajectory well, but the model will poorly predict cases from other initial conditions. To verify that this is not the case, for each of the 100 simulations above, an additional 30 simulations were performed from random initial conditions, using the optimized actual model in (5.2), the optimized expanded model in (5.3), and the continuous-time dynamics. The MSE of the trajectories from the actual discrete model and expanded models was aver-

aged over all 3,000 trajectories. The average MSE for the actual model was 83.9901, while the average MSE for the expanded model was 1.2017, indicating much better predictive capability and demonstrating that such models were indeed not overfit. Note that 13.87% of the actual models went unstable during these simulations (these were not included in the MSE calculations), while none of the expanded model trajectories went unstable.

These results illustrate that the CEM's tendency to produce false positives in the presence of unmodeled dynamics is actually caused by the fact that these additional model terms are useful in mitigating the effects of the model mismatch. When caused by time discretization errors, the tendency to produce false positives will be reduced by using a smaller sampling rate, or possibly by employing a higher-order method of time discretization.

5.1.5 Unmodeled Dynamics

Mathematical Understanding of Unmodeled Dynamics on the CEM

This section considers the case of unmodeled dynamics: the case where all the forces contributing to the systems dynamics whether due to modeling error/oversight or due to exogenous disturbances (i.e. turbulence). For this section, a linear mass spring system with continuous equations given in Equations (5.11-5.12) is considered.

$$\dot{x}_1 = x_2 \tag{5.11}$$

$$\dot{x}_2 = -\frac{k}{m}x_1 + Fsin(\omega t) \tag{5.12}$$

In order to add the sparsity to the system, the following discrete representation of the system given in Equation (5.13) is considered. T is the discrete time step and F is the magnitude

of the harmonic excitation.

$$\begin{bmatrix} x_1^{(t+1)} \\ x_2^{(t+1)} \end{bmatrix} = \begin{bmatrix} 1 & T & 0 & 0 & 0 \\ \frac{Tk}{m} & 1 & 0 & 0 & TF \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ cos(x_1) \\ sin(x_2) \\ sin(\omega t) \end{bmatrix} \tag{5.13}$$

Now, in order to explore unmodeled dynamics, the harmonic excitation term is removed from the model resulting in attempting to fit the model in Equation (5.14), which leaves the harmonic excitation as unmodeled dynamics.

$$\begin{bmatrix} x_1^{(t+1)} \\ x_2^{(t+1)} \end{bmatrix} = \begin{bmatrix} 1 & T & 0 & 0 \\ \frac{Tk}{m} & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ cos(x_1) \\ sin(x_2) \end{bmatrix} \tag{5.14}$$

The causation entropy can be written as a conditional mutual information as shown in [18]. However, the inclusion of conditioning on mutual information can function to either increase or decrease the mutual information [36]. Consider the causation entropy representation in terms of exclusively joint entropies as given in Equation (5.15).

$$C_{Z \to X|S} = H(X, S) + H(Z, S) - H(X, Z, S) - H(S) \tag{5.15}$$

For the CEM entries that would exist corresponding to Equation (5.14), the difference between including and not including the unmodeled dynamics is an appended entry to the S matrix that corresponds to the function of $sin(\omega t)$. Consider the S matrix appended with

the unmodeled dynamics as S^* in Equation (5.16) with S_1 being the unmodeled dynamics.

$$S^* = \begin{bmatrix} S & S_1 \end{bmatrix}^T \tag{5.16}$$

First the case where the unmodeled dynamics do not change the value of the Causation Entropy is considered.

In order of this to occur the condition of Equation (5.17) must be met.

$$C_{Z \to X|S} = C_{Z \to X|S^*} \tag{5.17}$$

Expanding Equation (5.17) gives the following condition.

$$\begin{gathered} H(X,S)+H(Z,S)-H(X,Z,S)-H(S) = H(X,S^*)+H(Z,S^*)-H(X,Z,S^*)-H(S^*) \\ H(X,S) + H(Z,S) - H(X,Z,S) - H(S) = \\ H(X, \begin{bmatrix} S & S_1 \end{bmatrix}) + H(Z, \begin{bmatrix} S & S_1 \end{bmatrix}) - H(X,Z, \begin{bmatrix} S & S_1 \end{bmatrix}) - H(\begin{bmatrix} S & S_1 \end{bmatrix}) \end{gathered} \tag{5.18}$$

In order for the final condition of Equation (5.18), the unmodeled dynamics, in this case $sin(\omega t)$ must be guaranteed independent of all entries in X, Z and S such that the joint entropies can be split and the unmodeled dynamics contribution to the entropies will cancel. However, this situation is impossible for the usage of the CEM for physical systems as the only way the unmodeled dynamics and X could be independent is if the unmodeled dynamics never contribute to the next time step of the dynamics, in which case the unmodeled dynamics are not actually dynamics of the system. Thus, the unmodeled dynamics must have an effect on the causation entropy value returned, but at first glance it is indeterminate what effect the unmodeled dynamics will have on the causation entropy values. In other words, the goal is to determine how $C_{Z \to X|S^*}$ is related to $C_{Z \to X|S}$. It turns out that unmodeled dynamics affect the CEM values in different ways, depending on whether $C_{Z \to X|S^*} > 0$ or $C_{Z \to X|S^*} = 0$.

Nonzero Causation Entropy Exploring the former case first, if $C_{Z\to X|S^*} > 0$, this implies that $H(X|S^*) > H(X|S^*, Z)$. Recall that $H(X|S^*)$ represents the amount of information required to describe X given S^*. Because all of the necessary covariates to describe the system dynamics are contained in the set $\{S^*, Z\}$, $H(X|S^*, Z) = 0$ because X is able to be fully determined given knowledge of S^* and Z. Thus, $C_{Z\to X|S^*} = H(X|S^*)$. Now, recall that $C_{Z\to X|S} = H(X|S) - H(X|S, Z)$. The quantity $H(X|S) < H(X|S^*)$ because less information is available from knowledge of S than knowledge of S^*. Likewise, $H(X|S, Z) > 0$ since the set $\{S, Z\}$ does not fully contain all the information needed to describe the system dynamics (and thus to be able to fully determine X). Thus, comparing the two causation entropy values yields,

$$\begin{aligned} C_{Z\to X|S^*} &= H(X|S^*) \\ &\geq H(X|S) - H(X|S, Z) \end{aligned} \tag{5.19}$$

Note that the (5.19) makes use of the fact that entropy values can never be negative. Recognizing the last term of (5.19) as $C_{Z\to X|S}$, it is observed that if the set of covariates does not include the necessary functions to describe the dynamic behavior, causation entropy values that are nonzero in the case of fully-modeled dynamics will decrease. The extent of this reduction in magnitude depends on the relative importance of the excluded terms in describing the dynamic behavior (this notion of sensitivity to specific model terms, and its effect on CEM values, was further explored in Section 3.2).

Zero Causation Entropy Consider the case when $C_{Z\to X|S^*} = 0$. In this case, the time history Z provides no information about X, above and beyond that already provided by knowledge of S^*. This means that $H(X|S^*) - H(X|S^*, Z) = 0$. However, if some data is removed from S^*, yielding the set of data S, it can no longer be guaranteed that knowledge of Z does not provide some additional information about X, because S no longer provides

Table 5.1: Table of results from unmodeled dynamics simulation

	Sparsity Acc.	False Pos.	False Neg.	Avg. False. Pos. Mag.
Full CEM	0.9930	0.070	0.00	0.1632
Reduced CEM	0.7825	1.74	0.00	0.1693

all information needed to determine X. Thus, it is possible that $H(X|S) > H(X|S,Z)$ and thus $C_{Z\to X|S} \neq 0$. This analysis shows that, while it is not guaranteed that a zero CEM entry (in the case of fully-modeled dynamics) becomes nonzero in the presence of unmodeled dynamics, it is possible if necessary covariates are removed. Whether or not a zero CEM entry becomes nonzero if a covariate is removed depends on whether that covariate was necessary to describe the dynamics of a particular state and on the nature of the still included, previously 0 covariates. The above effects of unmodeled dynamics on both zero and nonzero CEM entries will be further explored through several simulation examples below.

Simulation Results

In order to demonstrate these results, the system given in Equations (5.11-5.12) was simulated 100 times with random initial conditions for a total of 3 seconds each iteration. The full and reduced CEMs were computed in each case. The covariate selection accuracy of each is computed. Additionally, the average number of false positives and false negatives is recorded. In Table 5.1, the false positives (nonzero entry that should be zero) column is the average number of false positives per CEM computed. The false negatives column is the average number of false negatives (zero entry that should be nonzero). The average false positives column represents the average magnitude of the false positives that appear.

The results in Table 5.1 validate the expectations made in the sections above. First, the addition of unmodeled dynamics does indeed affect the results of the CEM. The appearance of false positives is far larger in the reduced case than the full CEM case; however, false negatives do not appear as discussed on the effects of unmodeled dynamics on nonzero

causation entropies. In the full CEM case, the occurrence of false positives was 0.07 false positives per CEM, or an incidence rate of 1.4% as there were 5 potential locations for a false positive to appear, while in the reduced case false positives had an occurrence of 1.74 false positives per CEM computed, which corresponds to an incidence rate of 43.5% as there were 4 potential entries for a false positive to appear.

Thus, it can be confidently concluded that if unmodeled dynamic do exist in a given system, this can lead to terms that would otherwise be expected to be zero to be nonzero (as they are needed to account for the unaccounted for portion of the generative dynamics); however, they fortunately will not cause for necessary terms to be incorrectly identified as zero as there were zero occurrences of necessary parameters being incorrectly identified as 0.

5.2 Considerations Stemming from Kernel Density Estimation

While the previous section explored general characteristics of CEM performance in the presence of noise independent of the probability estimation technique used, this section explores the impact of using kernel density estimation to compute the causation entropy values used in the CEM. First, a study on bandwidth selection in consideration of this problem is presented. Then, the effects of the dimension of the data and the so called curse of dimensionality on CEM estimation and performance is explored. Finally, there is a discussion on the implications of one of the underlying assumptions of using KDE: the assumption that the underlying PDF is stationary. This assumption affects the amount of data to include when attempting to calculate the CEM. A mathematical guide on how to select the best subset of data to include for causation entropy estimation is presented and followed up with a study on the effects of using a stationary PDF for a system and how informationless data can lead to poor PDF estimation.

5.2.1 Bandwidth Selection

In Section 2.2.1, a method for estimating PDFs through kernel density estimation was presented. Thus far, the one potential tuning parameter, the bandwidth, has been ignored with an expression provided on how to select it repeated here in Equation (5.20). This section explores the validity of Equation (5.20) for performing bandwidth selection for CEM computation in all case, including those with noise.

Gaussian KDE functions by placing a Gaussian at each data point and then summing the contributions of all the Gaussians at points where the composite PDF is desired. The bandwidth (h) describes the relationship between the width and height of the Gaussian and thus acts as a smoothing factor. A small bandwidth generates a skinny Gaussian which signifies large confidence in the data and may result in a non-smooth PDF when the distances between data points is large compared to the bandwidth. In Equation (5.20), d is the number of states and N the number of data points.

$$h = \left(\frac{4}{d+2}\right)^{\frac{1}{(d+4)}} (N)^{\frac{-1}{(d+4)}} \tag{5.20}$$

For studies done in this work, the bandwidth was varied by a scalar multiplier to adjust it in a meaningful way. A bandwidth multiplier of 1 correlates to the optimal bandwidth from [44]. The goal is to explore the connection between CEM estimator performance and bandwidth selection to hopefully validate a constant bandwidth multiplier expression independent of the level of noise present. If different amounts of noise require a different bandwidth selection, the proposed method will be far less useful as it is often difficult to quantify the exact amount of noise included in a data set. There is the potential that as the noise increases, the kernel size must appropriately increase in size in a corresponding manner in order to still encompass the true required state value.

In order to test this, measurement noise of varying magnitude was added to the measured states of an example system and the CEM computed using various bandwidths. The

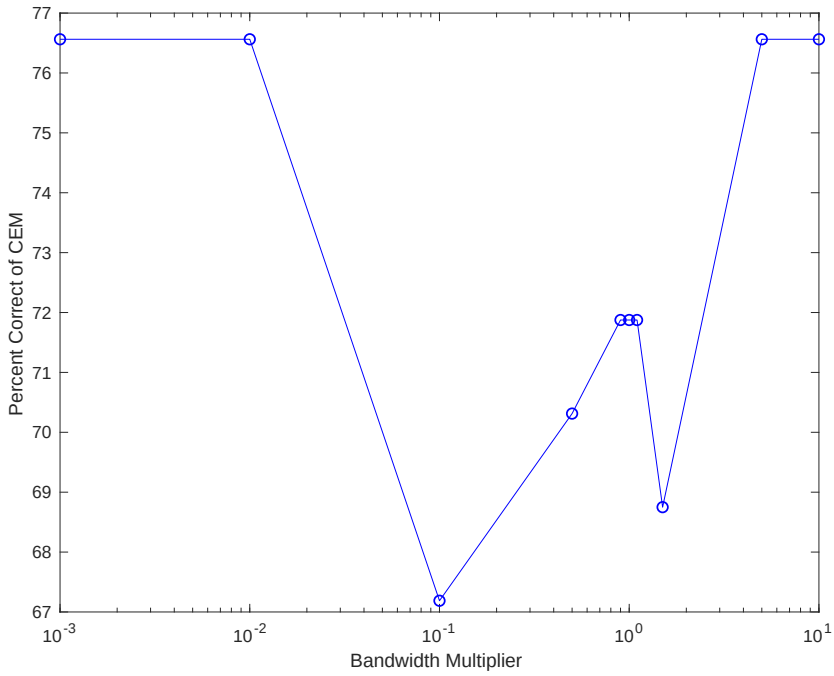

Figure 5.17: CEM Accuracy, Noise Multiplier $= 1e-5$

covariate selection accuracy of the CEM is defined as the percentage of cases which an entries structure is correct. An entry is correct if it is zero and should be zero or nonzero when it should be nonzero. An example of the study for a given level of noise is shown below in Figure 5.17. The figure demonstrates the accuracy of the CEM for various bandwidth multipliers for a given level of noise in a system. These results were then accumulated for various levels of noise with the results shown in Figure 5.18. The blue line represents the average optimal bandwidth and the red line the accuracy at the corresponding noise level. Based on these results, despite the bandwidth multipliers tested ranging from 10^{-3} to 10, the average optimal bandwidth always fell between 0.6 and 1.7 with it having little impact on the overall accuracy. Note that for this study the smallest bandwidth that provided the optimal CEM accuracy was selected as optimal, which implies that it is possible that a larger bandwidth multiplier could potentially generate an identically accurate CEM. Thus, using the optimal bandwidth from Equation (5.20) will be sufficient independent of the level of noise encountered, which is extremely beneficial for usage in system identification

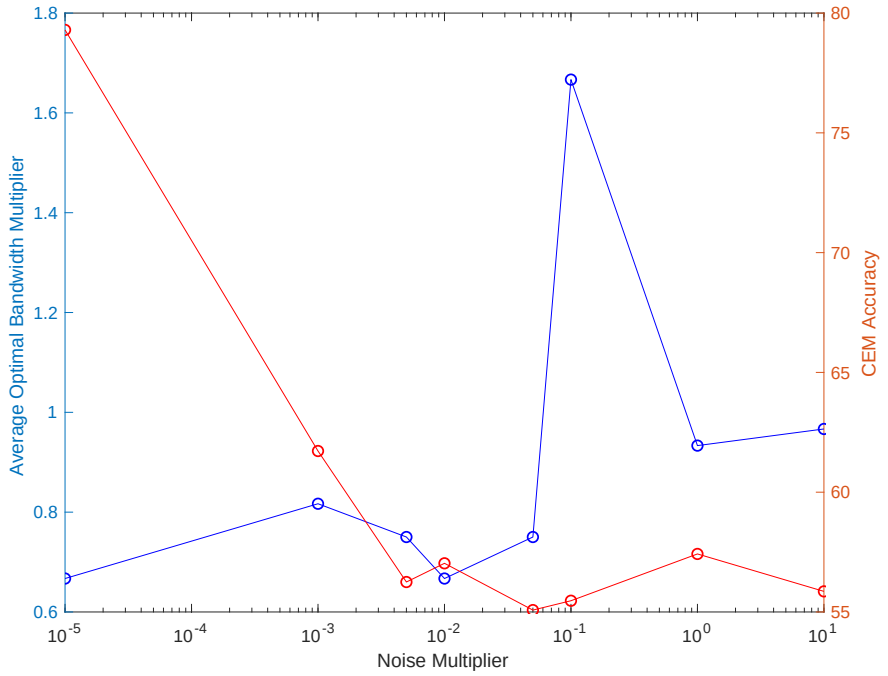

Figure 5.18: Optimal bandwidth multiplier, noise, and CEM accuracy

of unknown systems with potentially unknown noise magnitudes. This demonstrates that the observed results on the impact of measurement noise on the CEM is not simply a case of poor bandwidth selection but instead a deeper lying problem. Additionally, the results suggest that Equation (5.20) approximates the optimal bandwidth for estimating causation entropy.

The next Section 5.2.2, further explores the issues of bandwidth selection by demonstrating the problem of high dimensional Kernel Density Estimation.

5.2.2 Curse of Dimensionality and its Impact on CEM Estimation

This section explores the performance of the Causation Entropy Matrix as the dimension of the problem increases. From the definition of causation entropy in Equation (1.10), the computation of the causation entropy requires the value of $H(X_{t+1}|X_t, Y_t, Z_t)$, which necessitates the computation of $p(X_{t+1}, X_t, Y_t, Z_t)$. This calls for the estimation of a joint PDF that grows with the complexity of the system, particularly the number of functions

in the state function vector $\mathbf{F}(X_t, t)$ from Equation (1.11). It is known that KDE (and most PDF estimation methods) become increasingly more difficult as the dimension of the problem increases. In [74], the author suggests that KDE in up to $3 - 4$ dimensions has been successfully implemented, but with higher order usage becoming highly problematic. The cause of the failure in higher dimension is due to a phenomenon known as the curse of dimensionality. The curse of dimensionality comes down to bandwidth selection and choosing a neighborhood for each data point. If an appropriately small bandwidth is chosen so that only a small area around each data point is included, then the majority of the PDF space is empty. If sufficiently large neighborhoods around the data are included to leave the PDF non empty, then it is not an accurate neighborhood of the data and will lead to skewed results. Thus, the bias-variance trade-off is nearly impossible to sufficiently satisfy in higher dimension [43, 74]. A detailed discussion of the curse of dimensionality and its effects are documented in Chapter 7 Section 2 of [43].

In [56], extremely high accuracy of the CEM was shown for complex mechanical systems that could require PDF estimation of up to 13 dimensions. However, in [56], systems were run without noise or model mismatch. The re-substitution plugin estimator uses only observed data to generate and sample the PDF. In the case of perfect data, the PDF was accurate enough at the data points to provide accurate estimates of the causation entropy. This section explores the effects of the curse of dimensionality in the case of added measurement noise to the system.

In order to explore the effects of system dimension on the accuracy of the CEM, an experiment was run where linear systems with random sparsity structures in a series of dimensions from three to seven were generated and the corresponding model was propagated forward in time from random initial conditions. Gaussian measurement noise with a scalar noise multiplier n was then added to the system. Note that it was guaranteed that the system was marginally or asymptotically stable and that all of the states were unique and not identically equal to zero. If a random model violated one of the above conditions

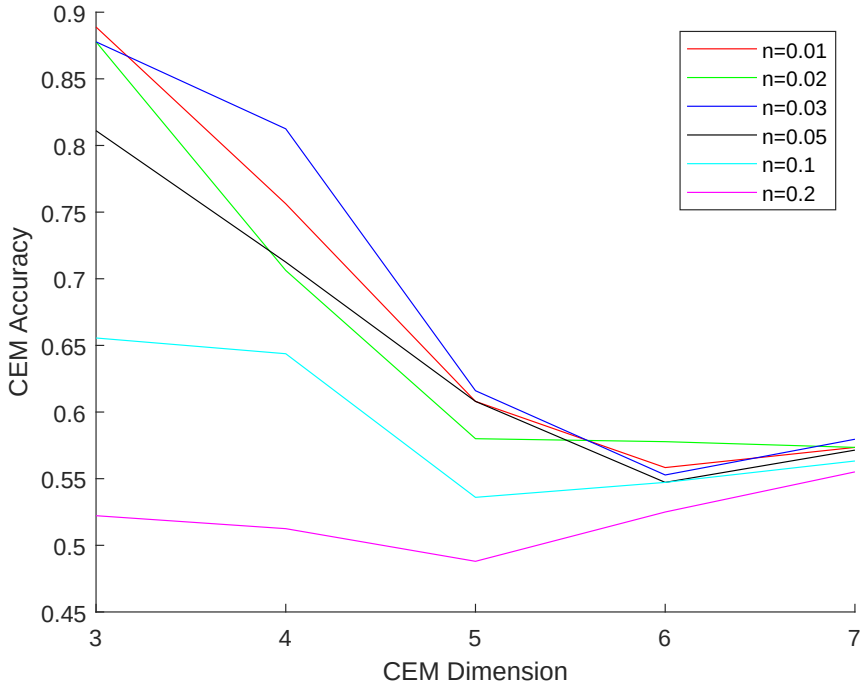

Figure 5.19: CEM accuracy vs dimension for various levels of noise

it was discarded and a new model generated. The CEM was then computed for the various systems. The experiment was run ten times with the results averaged and shown in Figures 5.19 and 5.20. Figure 5.19 demonstrates that for a given level of noise, as the dimension of the model increases, the accuracy of the CEM decreases. Note that the highest dimension PDF that needs to be estimated in the case of a linear system will be the dimension of the system plus one. Figure 5.20 demonstrates the average sparsity percentage (percentage of zeros) in the actual model. In order to guarantee that states were unique and not identically equal to zero, a number of nonzero entries are needed in the model. A higher percentage of nonzero entries are required in smaller dimension systems. Figure 5.20 demonstrates how as the dimension increases the average sparsity (occurrence of zeros) increased in the systems generated. In Section 5.1 it is shown that measurement noise drives the CEM to a zero matrix. Thus, the uptick at dimension 7 in Figure 5.19 is due to the fact that the sparsity percentage is increased, and especially in the case of large noise the matrix is mostly zeros and will thus have a greater CEM accuracy than in the case of smaller dimension sys-

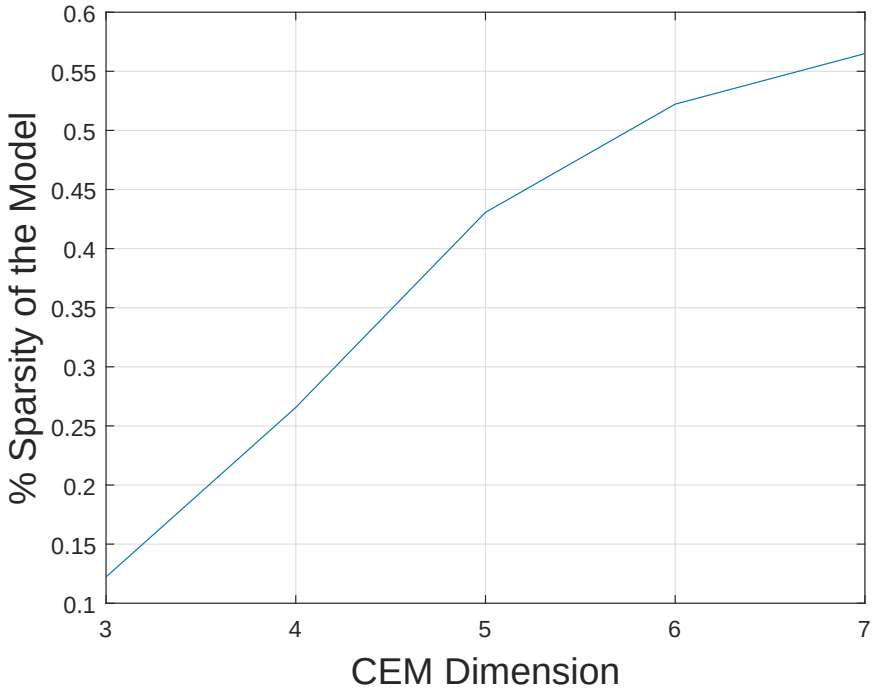

Figure 5.20: CEM sparsity percentage vs CEM dimension

tems. This result demonstrates that the curse of dimensionality appears when calculating the CEM. In the case of noiseless data, the KDE is accurate enough to allow the plugin estimator to accurately estimate the CEM. However, in the presence of noise, the lower the dimension of the system the more noise resistant and the higher the CEM accuracy will be.

5.2.3 Effects of Data Size

A natural question that arises when using the CEM for sparsity identification is the amount of data that should be used, and whether some portions of data should purposefully be included or excluded. As will be shown in this section, both the amount and type of data can affect the accuracy of CEM estimates, and thus the goal of the subsequent discussion is to provide practical guidance on how data should be selected when using the CEM.

Causation Entropy Upper Bound

The causation entropy values in the CEM are formulated to measure the causal connections between states in a dynamic system; however, the numerical values that the CEM entries take on is a function of the data used to estimate them. With sufficiently rich excitation of the dynamic system, the CEM produces accurate estimates of the system structure as observed in [56]. However, in the absence of dynamic excitation, a covariate function may appear to lack any connection to a state update even though such a connection might actually exist, leading erroneously to a zero causation entropy estimate. In practice, the causation entropy estimation scheme always produces estimates greater than zero, hence the need for the permutation test to determine whether such estimates should be statistically considered as zero. Thus, it is important to ensure that the causation entropy estimates are sufficiently high that the zero and non-zero entries can be statistically determined through the permutation test. The statistically-significant (non-zero) CEM values can be maximized through sufficient dynamic excitation, and also through intelligent selection of input data, to reduce the chances that a CEM entry will be erroneously classified as zero.

As a first step, an upper bound on causation entropy will be derived, leading to a useful technique for estimating the portions of data that should be included. The following theorem is offered to this end.

Theorem 1. *Let X and Z be continuous scalar random variables sampled at discrete times* $\{t, t+1, ...\}$. *Furthermore, let* $S = \{S_1, \ldots, S_N\}$ *be a set of continuous scalar random variables sampled at the same discrete times. Then,*

$$CE_{Z \to X|S} \leq H(X) \tag{5.21}$$

Proof. From the definitions of causation entropy in [18] and conditional entropy in [36],

the causation entropy in (5.21) can be rewritten as follows,

$$\begin{aligned} CE_{Z\to X|S} &= H(X|S) - H(X|S,Z) \\ &= H(X,S) + H(Z,S) - H(X,Z,S) - H(S) \end{aligned} \tag{5.22}$$

Also from [36], the following relationship exists between the joint and conditional entropies of multiple random variables (known as the entropy chain rule),

$$H(X,Y) = H(Y,X) = H(Y|X) + H(X) \geq H(X) \tag{5.23}$$

The inequality sign stems from the fact that entropy values are by definition non-negative. This property can easily be extended to the general case of N random variables. As a result of (5.19), Eq. (5.22) can be bounded as follows:

$$\begin{aligned} CE_{Z\to X|S} &\leq H(X,S) + H(Z,S) - H(Z,S) - H(S) \\ &\leq H(X,S) - H(S) \end{aligned} \tag{5.24}$$

Finally, it is well-known that joint entropy is subadditive [36], i.e., $H(X,S) \leq H(X) + H(S)$. Therefore, the inequality in (5.24) can be rewritten as,

$$\begin{aligned} CE_{Z\to X|S} &\leq H(X) + H(S) - H(S) \\ &\leq H(X) \end{aligned} \tag{5.25}$$

■

While it is not possible in general to derive a lower bound for causation entropy greater than zero, Theorem 1 does provide a means to calculate the upper bound. By choosing a subset of the measured data for X that maximizes $H(X)$, the associated causation entropy

$CE_{Z\to X|S}$ will have the highest possible upper bound. While this does not necessarily mean that $CE_{Z\to X|S}$ itself will be maximized, it does provide a general guideline for how to select the underlying data in a given problem.

Before showing some examples, it is important to note that many real-world problems involve multivariate datasets in which several states are measured, and several causation entropy values are computed (e.g., the entire CEM). Theoretically, for each state $X_i \in \{X_1, \ldots, X_n\}$ a particular time interval could be chosen for use in computing $CE_{Z\to X_i|S}$ that maximizes $H(X_i)$. In most practical problems, however, it is often desirable to choose a single time interval for all state data. In fact this use of a single time interval is usually required since the covariate functions in Eq. (1.11) are in general a function of all the states, and thus state data must be available to the estimator at common times.

To choose a single time interval over all states, consider again the subadditivity property for n random variables given by [36],

$$H(X_1, X_2, \ldots, X_n) \leq \sum_{i=1}^{n} H(X_i) \tag{5.26}$$

Suppose that a time interval $[t_0, t_f]$ is identified that maximizes $H(X_1, X_2, \ldots, X_n)$ over all of the available data. Then the individual entropy values $H(X_i)$ will be at least as large as the joint entropy. Thus, a general guideline for selecting the data to include in CEM estimation is that which maximizes the joint entropy of the system states. By maximizing the joint entropy, the individual entropies of each state will be lower-bounded by this value, and by

Because the individual entriopies are lower-bounded by the joint entropy, by maximizing the joint entropy the upper bound on the causation entropies are also maximized according to Theorem 1.

Thus, to improve the statistical significance of the values in the CEM and improve error characteristics, a general guideline is to select a data sequence that maximizes the joint entropy of the states. This will be demonstrated through an example below. Note that,

while the above discussion refers to one data sequence over a single time interval $[t_0, t_f]$, it is possible to combine data over multiple time intervals to further increase the joint entropy, although such a development is not explored here.

Simulation Examples Two simulation examples are presented in this section to clarify and illustrate the overall dependence of the CEM accuracy on the selected interval data. The first example illustrates how the underlying PDF estimates produced by KDE change as data from different portions of a timeseries are included. The second example extends this concept to illustrate how the magnitudes of the CEM values and the CEM accuracy changes as a function of the interval of data selected.

To begin with, consider an autonomous, stable, second-order linear system given by,

$$\dot{\mathbf{x}} = \begin{bmatrix} -1.05 & 0.75 \\ -2.5 & 0.25 \end{bmatrix} \mathbf{x} \tag{5.27}$$

To illustrate the behavior of the joint PDF of the states estimated from discrete data samples, the system is integrated forward in time from initial condition $\mathbf{x}(t = 0) = \{X, X\}^T$ for a period of 500 sec using a sampling time of 0.05 sec. Using this complete data sequence, six subsequences are selected for evaluation. These data subsequences correspond to the first 0.45 sec of data, the first 2.5 sec, the first 5 sec, the first 10 sec, the first 100 sec, and the entire 500 sec of data. The time histories of the states for each of these subsequences are shown in Fig. 5.21 (left). The entire 500 sec can be divided into transient (approximately first 10 sec) and steady state portions, as shown in the figure.

The joint PDFs of the states for each of these subsequences constructed using the KDE approach in the previous section are shown in the right of Fig. 5.21. It is interesting to observe how the PDF changes over time. Using only the first 0.45 sec of data, the correlated nature of the state evolution is evident in the PDF, but incomplete. Once the bulk of the transient response is included in the data sequence (2^{nd}, 3^{rd}, and 4^{th} subplots on the right of

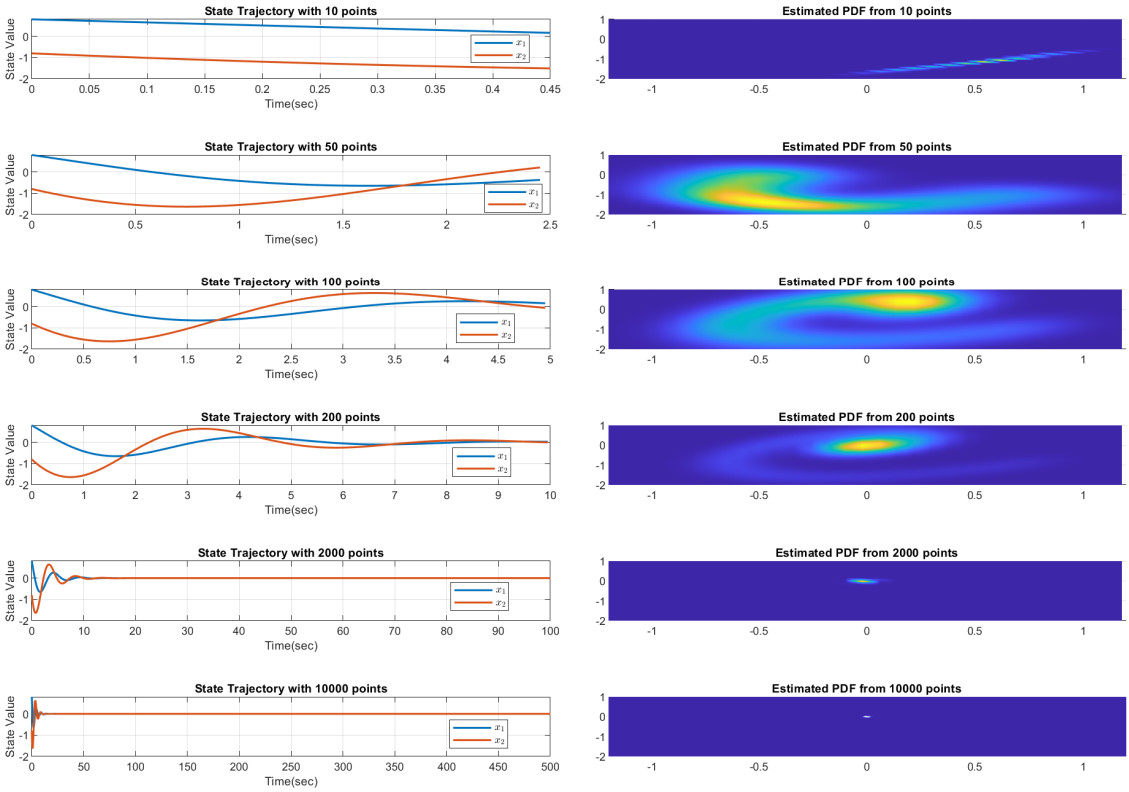

Figure 5.21: Trajectory and Corresponding PDF

Fig. 5.21, the fully correlated and information-rich PDF produced by the density estimator. However, the bottom two subplots on the right of Fig. 5.21 show what happens when a large portion of the steady-state response is included – in this case, the PDF approaches a delta function around the steady-state values. This is because the steady-state time series essentially no information about the dynamics of the systems, and in fact the joint entropy of the steady-state portion of the timeseries is zero. The approximate delta function PDF produced from the full 500 sec time series is undesirable from a CEM estimation standpoint as it contains little information about the underlying between the states and the covariates.

One can view the results in Fig. 5.21 as an interesting, and potentially problematic, outcome of applying information-theoretic quantities derived for random variables to data generated from a deterministic dynamical system. The PDF generated from variable-length data sequences produced from a dynamic system are non-stationary in general – depending on the time interval selected to observe the states, the PDF will change [75]. Because causation entropy is estimated directly from the joint PDF of the states, its value will vary depending on the portion of data selected as mentioned earlier. This non-stationary property of the underlying PDF is important to recognize in practical application of CEM sparsity estimation as it explains why proper selection of the input data sequence is so important.

To further illustrate this, consider now a system of two masses coupled by nonlinear springs and dampers. The system is connected to a rigid wall at one end. The equations of motion for each mass are given by,

$$\begin{aligned} \ddot{x}_1 &= \frac{1}{m_1}[-f_s^{(1)}(x_1) - f_d^{(1)}(\dot{x}_1) + f_s^{(2)}(x_2 - x_1) + f_d^{(2)}(\dot{x}_2 - \dot{x}_1)] \\ \ddot{x}_2 &= \frac{1}{m_2}[-f_s^{(2)}(x_2 - x_1) - f_d^{(2)}(\dot{x}_2 - \dot{x}_1)] \end{aligned} \tag{5.28}$$

where the spring and damper forces between each mass are given respectively by,

$$f_s^{(i)}(x) = k_i x + b_i x^3 \tag{5.29}$$

$$f_d^{(i)}(\dot{x}) = c_i \dot{x} \tag{5.30}$$

In this work, the following parameter values are used: $m_1 = m_2 = 1$ kg, $k_1 = 20$ N/m, $k_2 = 10$ N/m, $b_1 = 5$ N/m^3, $b_2 = 10$ N/m^3, $c_1 = 2$ Ns/m and $c_2 = 4.5$ Ns/m. The system in (5.28) can be written in first-order form, discretized using a zero-order-hold transform with a time step of 0.01 sec, and written in the form of Eq. (1.11) [56]. The resulting four-state system has an 4×10 parameter matrix Θ with 16 non-zero entries.

To illustrate the effect of data selection on CEM accuracy, a set of five trajectories are generated from five random initial conditions and simulated for 20 sec. One of these trajectories is shown in Fig. 5.22. Since the masses are unforced, the system trajectories exhibit a transient portion followed by a steady-state period once the masses reach their equilibrium. In this example, these trajectories are analyzed first with no noise added, and also with zero-mean Gaussian noise added to the state time histories (where the noise standard deviation is 0.001).

For each state time history, the joint entropy of the four states is computed using subsequences of various data length (the total data sequence has 2,000 samples for each time history). The joint entropies for each data length are then averaged over the five trajectories. The top plot in Fig. 5.23 shows the average joint entropy as a function of data length for the no-noise case, while the bottom shows the average joint entropy when noise is included in the measurements. Note that the joint entropy is maximized in both cases during the initial transient, and then decays as data with low information content from the steady-state response is included. This is a direct result of the effect shown in Fig. 5.21 – at the states evolve, the PDF derived from the time histories changes from one of high joint entropy to one of low joint entropy as the steady-state response begins to make up the dominant portion of the time series.

The changing values of the joint entropy directly affect the causation entropy estimates. Figures 5.24 and 5.23 show the average magnitude of the nonzero entries in the CEM, and

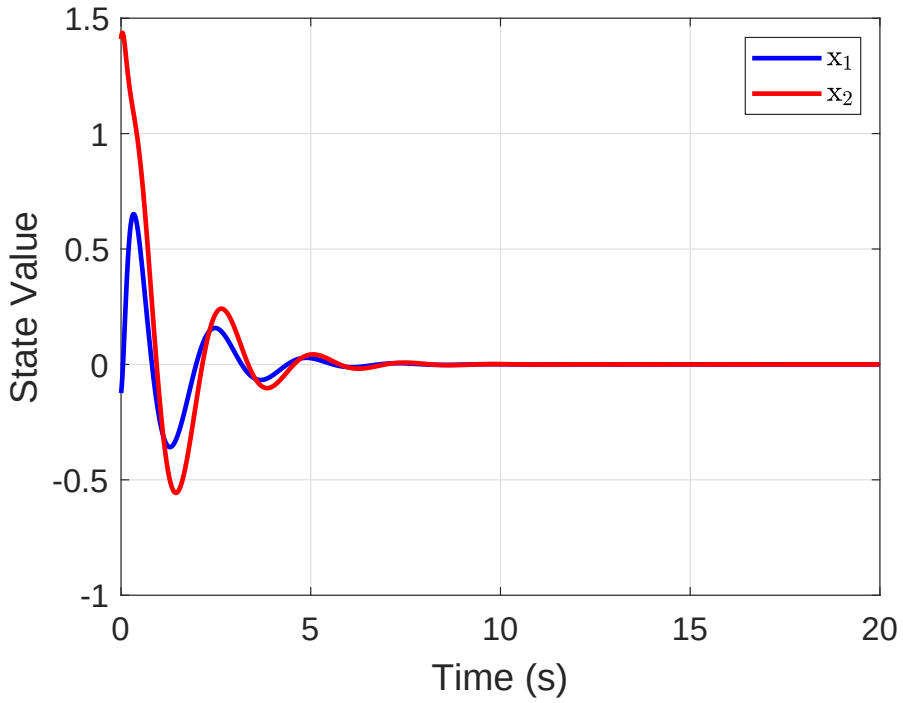

Figure 5.22: Sample Trajectory of Oscillator Masses

the CEM accuracy (as defined in the previous section), respectively. In each figure, the results for the no-noise case are shown in the top plot, and for the noisy case in the bottom plot. Each value is calculated as the average over the five trials, for variable data length. In the case of perfect data, the CEM magnitudes reach their maximum at approximately the same data length that maximizes the joint entropy, and then remain at roughly the same average magnitude thereafter. This robustness to low-information data is actually a result of the plugin estimator as it both generates then samples the relevant PDF solely at locations where data was encountered. Regardless of the amount of informationless data, there will be support of some (albeit potentially minimal) size to sample at the observed, true data point. Thus, if there is no measurement noise or model mismatch, the sampled data point directly demonstrates the generative dynamics' causal relationship. In Section 2.2.3, it was demonstrated that the scaling of the PDF value is irrelevant; only the relative magnitudes at a given point between PDFs used for CEM computation matter; even when the support is skewed by informationless data, the relative magnitudes at individual points

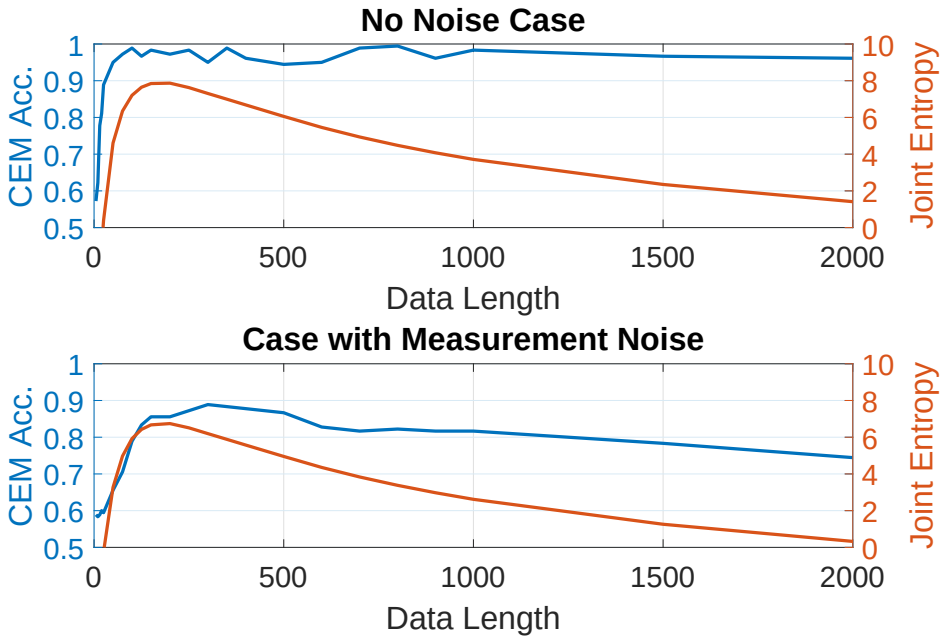

Figure 5.23: Comparison of CEM covariate selection accuracy and joint entropy as functions of data length in no noise and measurement noise cases.

are maintained. Thus, there is no decay in the CEM magnitude in the case of perfect data regardless of the amount of data included.However, even in the presence of a small amount of measurement noise, the average CEM magnitude and the CEM accuracy both degrade as more steady-state data is included, as shown in the bottom plots of Figs. 5.24 and 5.23. As low-information data makes up a greater proportion of the dataset, the PDF converges to a delta function at the origin, reducing the support around all non-origin data points as well as the joint entropy. This has the effect of decreasing the causation entropy values. CEM entries which are low to begin with (due to weak, but nonzero causal influence due to decreased support at the true causal location and associated to lower parameter values) are then harder to differentiate from entries that should statistically be considered zero, and thus are erroneously eliminated via the permutation test. This results in an overall reduction in accuracy as more of the steady-state portion of the data is included.

The results of this section have several implications. First, the joint entropy of the data sequence is a useful metric in determining what portion of data should be selected when

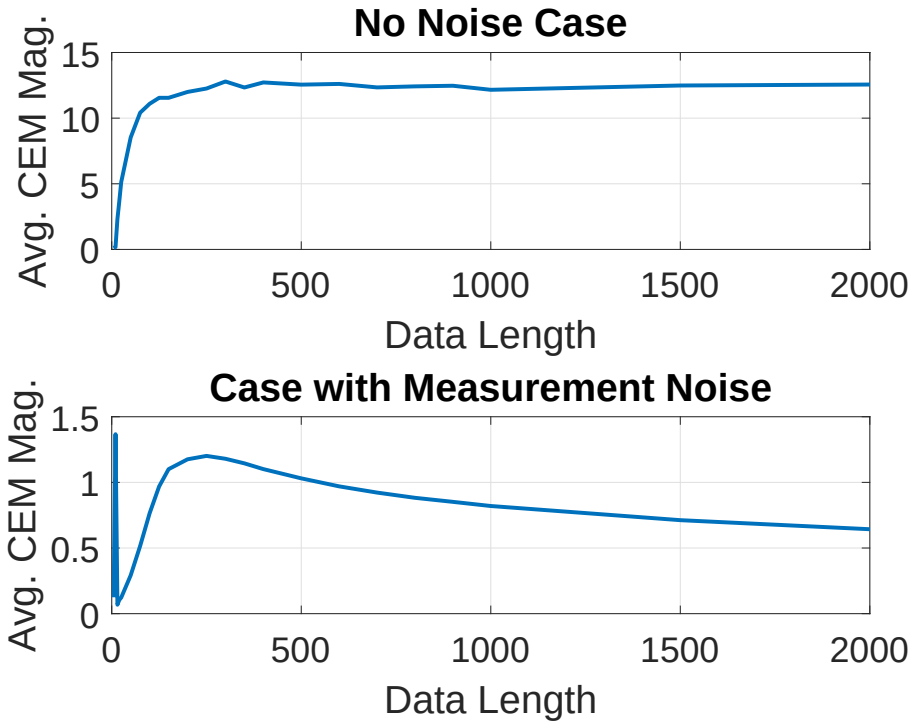

Figure 5.24: Average CEM magnitude vs data length for no noise and measurement noise cases.

computing the CEM. Generally speaking, selecting the portion of data that maximizes the joint entropy will lead to higher overall CEM values and higher overall accuracy, since statistically zero and nonzero CEM values can be better differentiated. Second, the cases in this section were limited to unactuated dynamic systems where the steady-state response provides essentially no information about the dynamics of the system. In cases involving continual excitation of the dynamics, either through control inputs or external disturbances (e.g., several of the examples in [56], proper data selection may not be so critical as the entire time history may be information-rich, leading to high joint entropy values regardless of which portion of the data is selected. Finally, while the no-noise case shown here was robust to the inclusion of data with low information content, in real-world applications such perfect data is typically not available, and thus the joint entropy-based data selection technique described here should be applied in the absence of persistent excitation of the dynamics.

CHAPTER 6
PHYSICAL SYSTEM EXPERIMENTATION

This chapter details the experimental validation of the Causation Entropy Matrix through its application to a physical, nonlinear system. The system studied is a ball rolling back and forth atop a table that pivots like a seesaw. This chapter proceeds by first describing the experimental setup used and deriving the dynamics of the system from first principles. Subsequently, the applicability of the idealized model is then experimentally validated. Finally results of the CEM computation for the system and the performance of the optimized model are provided with a corresponding discussion.

6.1 Experimental Setup

In order to test the proposed system, a setup was made with a track for a ball to roll along where the angular displacement of the table and the linear displacement of the ball are recorded. The goal of the experiment is offline system identification, so all derivatives of the states and inputs can be calculated and used for model fitting. Figures 6.1 and 6.2 show the experimental testbed used in this system. The black box on the left side of the setup in Figure 6.1 is a Garmin Lidar Lite V4 laser range finder with a stated accuracy of $\pm$ 1 cm. Visible at the top of Figure 6.2 is a Dynamixel MX-28 servo that has its torque disengaged and functioned passively as an absolute encoder. It has a stated accuracy of 0.088 degrees. The length of the track from tip of the Garmin sensor to the stop at the far end of the track is 90 cm. In all derivations, a flat track is assumed in order to simplify the dynamics. Here, two rails are used to minimize any sideways motion of the ball. The gap between the two rails is 4 cm. The rails contact the ball near its bottom to approximate rolling on a flat surface. Inputs were manually applied to the table to keep the ball rolling without hitting the ends of the rail while maintaining varied motion of the ball. The angular displacement

of the table θ (and its derivatives) are considered known control inputs.

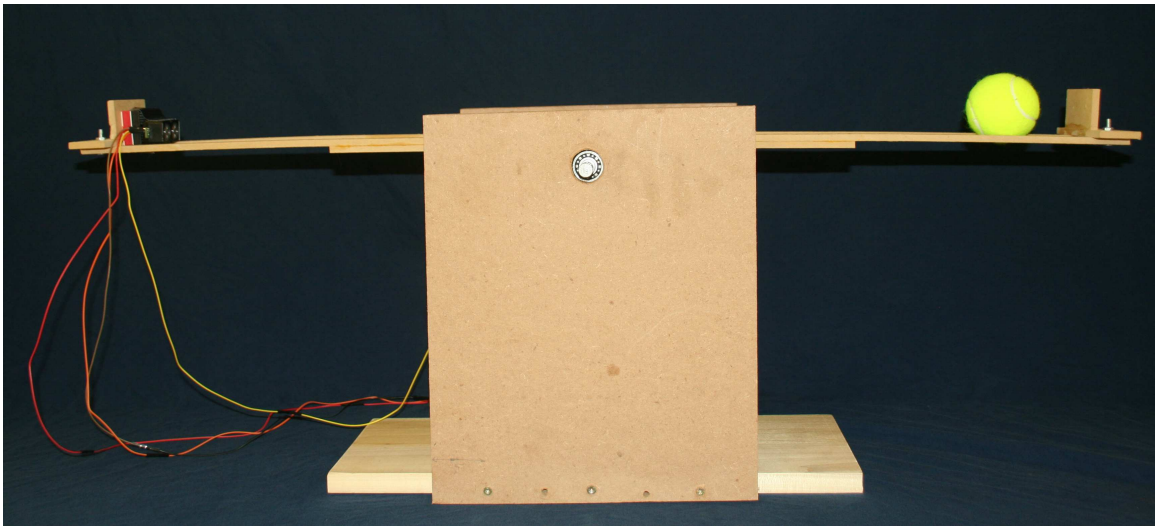

Figure 6.1: Side View of experimental system

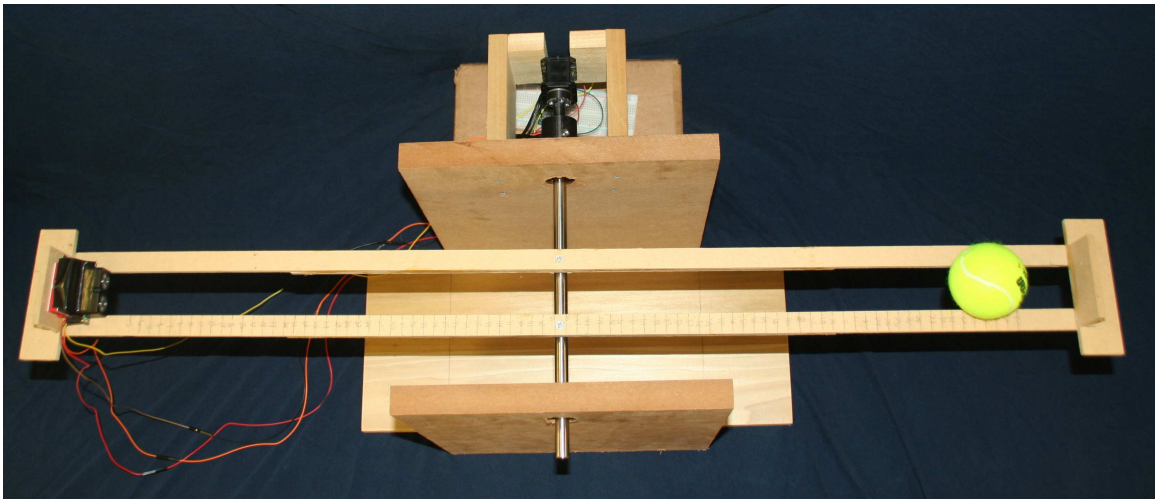

Figure 6.2: Top down view of experimental system

6.2 Kinematics and Dynamics of Physical System

In order to be able to accurately quantify the performance of the CEM and perform accurate modeling, the dynamics of the system must be derived. The system studied is defined for modeling purposes as shown in Figure 6.3.

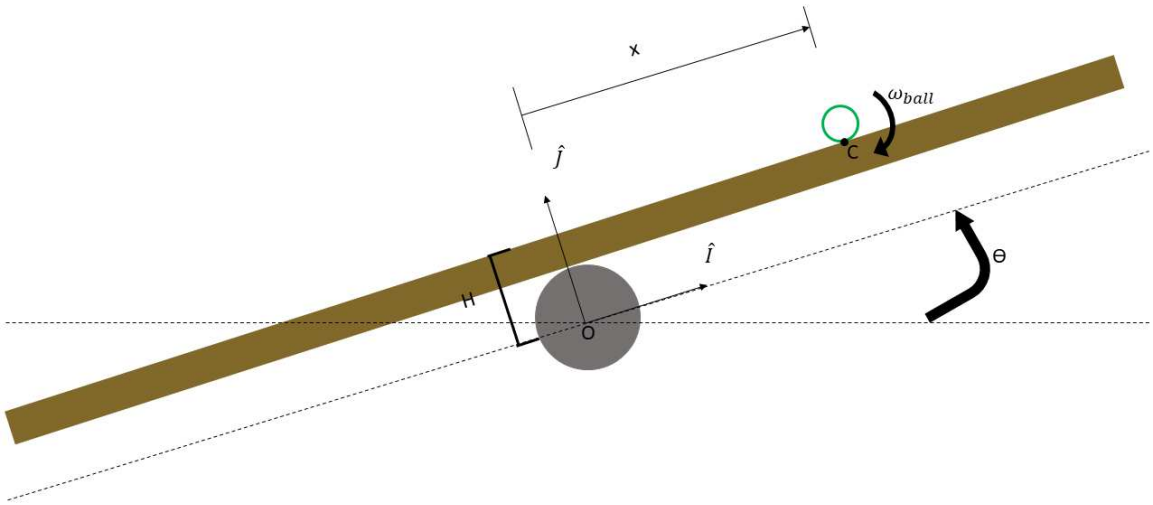

Figure 6.3: Model of physical system

6.2.1 Kinematics of the Rolling Ball

For the system studied, it is assumed that the ball rolls without slipping. The rolling without slip condition requires that there is no relative motion parallel to the contact plane [76]. In this case, the relative motion of the ball in the direction parallel to the plane of rolling is based on changes in the position $\dot{x}\hat{i}$. This condition defines the angular velocity ω of the ball itself as shown below in Equation (6.1).

$$\begin{aligned} (v_C)^{ijk} &= (v_{cm})^{ijk} + \omega \times r_{c/cm} = 0 \\ \dot{x}\hat{i} + (-\omega_{ball}\hat{k} \times -r\hat{j}) &= 0 \\ (\dot{x} - \omega_{ball})\hat{i} &= 0 \\ \omega_{ball} &= \frac{\dot{x}}{r} \end{aligned} \tag{6.1}$$

6.2.2 Newton Euler Derivation

The results of Equation (6.1) are useful in defining the total angular velocity and total angular acceleration of the ball ω_{total} and α_{total}. First, ω_{total} is defined in Equation (6.2).

$$\omega_{total} = (\dot{\theta} - \omega_{ball})\hat{k} \tag{6.2}$$

The relationship in Equation (6.2) is valid for all time, and thus may be differentiated to yield an expression for α_{total} as shown in Equation (6.3).

$$\begin{aligned}
\alpha_{total} &= \frac{d}{dt}\omega_{total} \\
\alpha_{total} &= \frac{d}{dt}[(\dot{\theta} - \omega_{ball})\hat{k}] \\
\alpha_{total} &= (\ddot{\theta} - \frac{d}{dt}\frac{\dot{x}}{r})\hat{k} + (\dot{\theta} - \omega_{ball})\dot{\hat{k}} \\
\alpha_{total} &= (\ddot{\theta} - \frac{\ddot{x}}{r})\hat{k}
\end{aligned} \tag{6.3}$$

In the above case $\dot{\hat{k}}$ is equal to zero as there is no change over time in the $\hat{k}$ axis. The equations of motion can be defined by writing the sums of forces and moments as given in Equations (6.4- 6.6). A free body diagram of the ball is given in Figure 6.4.

$$\sum F_x : -f_f - mg\sin(\theta) = m(a_{cm} \cdot \hat{i}) \tag{6.4}$$

$$\sum F_y : N - mg\cos(\theta) = m(a_{cm} \cdot \hat{j}) \tag{6.5}$$

$$\sum \tau_{cm} : -r\hat{j} \times -f_f\hat{i} = I_{zz}^{cm}\alpha_{total} \cdot \hat{k}$$

$$\sum \tau_{cm} : -f_f r = I_{zz}^{cm}\alpha_{total} \tag{6.6}$$

Equation (6.6) can be rearranged to provide an explicit equation for the force of friction f_f exerted on the ball, which can then be substituted into Equation (6.4) to remove the unknown.

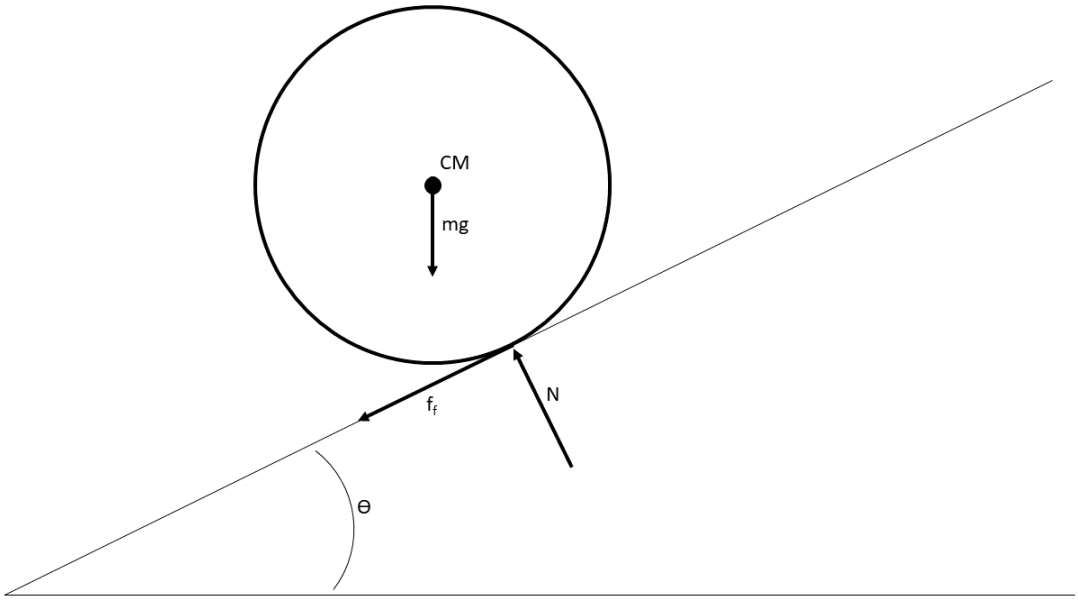

Figure 6.4: Free body diagram of the system's rolling ball

In order to have a solvable expression for equation of motion of the ball, the acceleration of the center of the mass of the ball a_{cm} must be known. The acceleration can be derived by considering a point accelerating in a moving reference frame as given in [76].

$$a_{cm} = a_O + (a_{cm})^{ijk} + \ddot{\theta} \times r_{cm/O} + \dot{\theta} \times (\dot{\theta} \times r_{cm/O}) + 2\dot{\theta} \times (v_{cm})^{ijk} \tag{6.7}$$

For the system studied, $r_{cm/O}$ is the distance from the center of the pivot shaft to the center of mass of the ball, which is given in Equation (6.8), with Δz defined for convenience.

$$\begin{aligned} r_{cm/O} &= x\hat{i} + (h + r)\hat{j} \\ r_{cm/O} &= x\hat{i} + \Delta z\hat{j} \end{aligned} \tag{6.8}$$

Thus, all terms in Equation (6.7) are known and can be evaluated as shown in Equation (6.9).

$$a_{cm} = (\ddot{x} - \ddot{\theta}\Delta z - \dot{\theta}^2 x)\hat{i} + (\ddot{\theta}x + 2\dot{\theta}\dot{x} - \dot{\theta}^2\Delta z)\hat{j} \tag{6.9}$$

Thus, the equation of motion of the ball follows. Substituting into Equation (6.6) yields:

$$\begin{aligned} -f_f r &= I_{zz}^{cm}(\alpha_{total}) \\ -f_f r &= I_{zz}^{cm}(\ddot{\theta} - \frac{\ddot{x}}{r}) \\ -f_f &= \frac{I_{zz}^{cm}}{r}(\ddot{\theta} - \frac{\ddot{x}}{r}) \end{aligned} \tag{6.10}$$

Substituting for f_f into Equation (6.4) yields the final equation of motion given in Equation (6.11).

$$\begin{aligned} -f_f - mg\sin(\theta) &= m(a_{cm} \cdot \hat{i}) \\ \frac{I_{zz}^{cm}}{r}(\ddot{\theta} - \frac{\ddot{x}}{r}) - mg\sin(\theta) &= m(\ddot{x} - \ddot{\theta}\Delta z - \dot{\theta}^2 x) \end{aligned} \tag{6.11}$$

Thus, the simplified final equation of motion is given in Equation (6.12).

$$m\ddot{x} - \frac{I_{zz}^{cm}}{r}\ddot{\theta} + \frac{I_{zz}^{cm}}{r^2}\ddot{x} + mg\sin(\theta) - m\dot{\theta}^2 x - m\ddot{\theta}\Delta z = 0 \tag{6.12}$$

6.2.3 Lagrange's Equation Derivation

In order to verify the result shown in Equation (6.12), a derivation through Lagrange's equation was performed. The total kinetic energy T can be defined as given in Equation (6.13) [76].

$$T = \frac{1}{2}mv_{cm} \cdot v_{cm} + \frac{1}{2}\omega \cdot H_{cm} \tag{6.13}$$

H_g is the angular momentum about g. In the planar motion case with with a symmetric body and the center of mass considered, H_{cm} is $I_{zz}^{cm}\omega$. In order to define the kinetic energy T, v_{cm} is required.

$$\begin{aligned} v_{cm} &= v_O + (v_{cm})^{ijk} + \dot{\theta} \times r_{cm/O} \\ v_{cm} &= (\dot{x} - \dot{\theta}\Delta z)\hat{i} + \dot{\theta}x\hat{j} \end{aligned} \tag{6.14}$$

Thus, T can be defined as in Equation (6.15) using ω_{total} from Equation (6.2) for ω in Equation (6.13).

$$T = \frac{1}{2}m\left((\dot{x} - \dot{\theta}\Delta z)^2 + \dot{\theta}^2 x^2\right) + \frac{1}{2}I_{zz}^{cm}(\dot{\theta} - \frac{\dot{x}}{r})^2 \tag{6.15}$$

The potential energy V of the ball can be defined as given below in Equation (6.16) when using O as a datum.

$$V = mg\left(x\sin(\theta) + \Delta z\cos(\theta)\right) \tag{6.16}$$

With these quantities, Lagrange's equations, given by Equation (6.17), can be used. In this case, there are no generalized forces Q applied to the system, and as the assumption of rolling without slipping is made and no work is done by a non conservative force, Lagrange's equations may be used with no modification.

$$\begin{aligned} L &= T - V \\ \frac{d}{dt}(\frac{\partial L}{\partial \dot{x}}) - \frac{\partial L}{\partial x} &= Q \end{aligned} \tag{6.17}$$

Evaluating the components of Equation (6.17) can be evaluated as below in Equation (6.18).

$$\begin{aligned} \frac{d}{dt}(\frac{\partial L}{\partial x}) &= m\ddot{x} - \frac{I_{zz}^{cm}}{r}\ddot{\theta} + \frac{I_{zz}^{cm}}{r^2}\ddot{x} - m\ddot{\theta}\Delta z \\ \frac{\partial L}{\partial x} &= mx\dot{\theta}^2 - mg\sin(\theta) \end{aligned} \tag{6.18}$$

Combining the components of Equation (6.18) yields the complete equation of motion shown below in Equation (6.19).

$$m\ddot{x} - \frac{I_{zz}^{cm}}{r}\ddot{\theta} + \frac{I_{zz}^{cm}}{r^2}\ddot{x} - m\ddot{\theta}\Delta z + mg\sin(\theta) - mx\dot{\theta}^2 = 0 \tag{6.19}$$

Notice that Equations (6.12) and (6.19) are identical despite being solved through separate methods. Thus, the included expression defines the idealized motion of the of the ball.

6.3 Discrete Model Representation

Using the above equation of motion from Equation (6.19), the minimal CEM representation can be generated. First, combining and similar terms generates a simplified representation as shown below in Equation (6.20).

$$(m + \frac{I_{zz}^{cm}}{r^2})\ddot{x} - (m\Delta z + \frac{I_{zz}^{cm}}{r})\ddot{\theta} + m\dot{\theta}^2 x - mg\sin(\theta) = 0 \tag{6.20}$$

Equation (6.20) can be written more concisely if by defining γ and ζ as below in Equation (6.21).

$$\begin{aligned} \gamma &= m + \frac{I_{zz}^{cm}}{r^2} \\ \zeta &= m\Delta z + \frac{I_{zz}^{cm}}{r} \end{aligned} \tag{6.21}$$

Substituting the values from Equation (6.21) into Equation (6.20) and solving for $\ddot{x}$ yields Equation (6.22), which is the simplest representation of the EOM.

$$\ddot{x} = \frac{\zeta}{\gamma}\ddot{\theta} + \frac{m}{\gamma}\dot{\theta}^2 x - \frac{mg}{\gamma}\sin(\theta) \tag{6.22}$$

This equation can be decomposed into a pair of first order ordinary differential equations using the transformation $x_1 = x$ and $x_2 = \dot{x}$ given in Equations (6.23-6.24).

$$\dot{x_1} = x_2 \tag{6.23}$$

$$\dot{x_2} = \frac{\zeta}{\gamma}\ddot{\theta} + \frac{m}{\gamma}\dot{\theta}^2 x_1 - \frac{mg}{\gamma}\sin(\theta) \tag{6.24}$$

Equations (6.23-6.24) can the be discretized yielding Equation (6.25) where $[\cdot]_t$ represents

the value of $\cdot$ evaluated at time t.

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{t+1} = \begin{bmatrix} Tx_2 + x_1 \\ \left(T * \left(\frac{\zeta}{\gamma}\ddot{\theta} + \frac{m}{\gamma}\dot{\theta}^2 x_1 - \frac{mg}{\gamma}\sin(\theta) \right) + x_2 \right) \end{bmatrix}_t \tag{6.25}$$

Thus, the minimal represenation of the dynamics can be written in the form of (1.11) as shown below in Equation (6.26).

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_{t+1} = \begin{bmatrix} 1 & T & 0 & 0 & 0 \\ 0 & 1 & \frac{T\zeta}{\gamma} & \frac{Tm}{\gamma} & -\frac{Tmg}{\gamma} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \ddot{\theta} \\ \dot{\theta}^2 x_1 \\ \sin\theta \end{bmatrix}_t \tag{6.26}$$

Thus, assuming the no-slip condition of the ball is met and a sufficiently small time step is used, Equation (6.26) provides a minimal representation of the discrete dynamics of the system. Ideally the CEM should model the structure of the parameter matrix Θ in Equation (6.26). Models containing no prior knowledge of the generative dynamics structure where every parameter is potentially nonzero will be referred to as Model 1 for the remainder of this work. The idealized dynamics model structure contained in equation (6.26) will be referred to as Model 2 for the remainder of this work. In this work the structure of a model is the set of included functions (and where they appear), which is analogous to the location of zero verses nonzero entries in the corresponding Θ matrix.

6.4 Experimental Methodology

The remainder of this chapter seeks to validate both the validity of the derived model as well as demonstrate the successful application of the CEM to experimentally collected data. This section will cover the data collected as well as the experimental procedure used to generate the results.

6.4.1 Data Collected

Two distinct, independent, experimental data sets were collected. In both cases, the table was manually excited to provide a rich input set, which kept the ball having varied movements while ensuring that it never hit the edge of the track. The experimentally collected data sets will be referred to as Data Set 1 and Data Set 2 for the remainder of this work. Plots of the filtered trajectories are given in Figure 6.5. Data Set 1 was collected with an average time step of 0.003 seconds, a minimum time step 0.002 seconds, a maximum time step of 0.004 seconds, and a time step standard deviation of $7.6362 * 10^{-4}$ seconds. Data Set 2 had the same mean, min and max time step values as the training set; however, it had a standard deviation of $5.6034 * 10^{-4}$ seconds. Thus, the time step for both the training and validation sets very nearly approaches a constant time step of 0.003 seconds. The slight variation in timing occurs due to slight variations in the motor's performance and communication as the data collection was run by polling the sensors at the highest sampling rate possible, which accounts for the slight variations in time step per data point.

The next section details the necessary steps to proceed from the collected sensor data to the completed and displayed trajectories.

6.4.2 Data Transformation

The data generated by the sensors provides the distance from the leading edge of the ball to the laser range finder and the absolute angular position in degrees of the servo shaft. However, in order to use the model derived, the axis must be body fixed at the pivot with the positive $\hat{i}$ direction considered towards the laser range finder. First, in order to transform the table angle measurements, a reading of the table angle was taken when level; this value was then subtracted from the measurement reading at each time step to match the $\hat{i}\hat{j}\hat{k}$ frame (note that special care has to be taken to ensure that positive θ corresponds to a positive rotation about $\hat{k}$.) In order to transform measured position sensor data into state data, define l as the length from the Garmin sensor edge to the opposite end of the table,

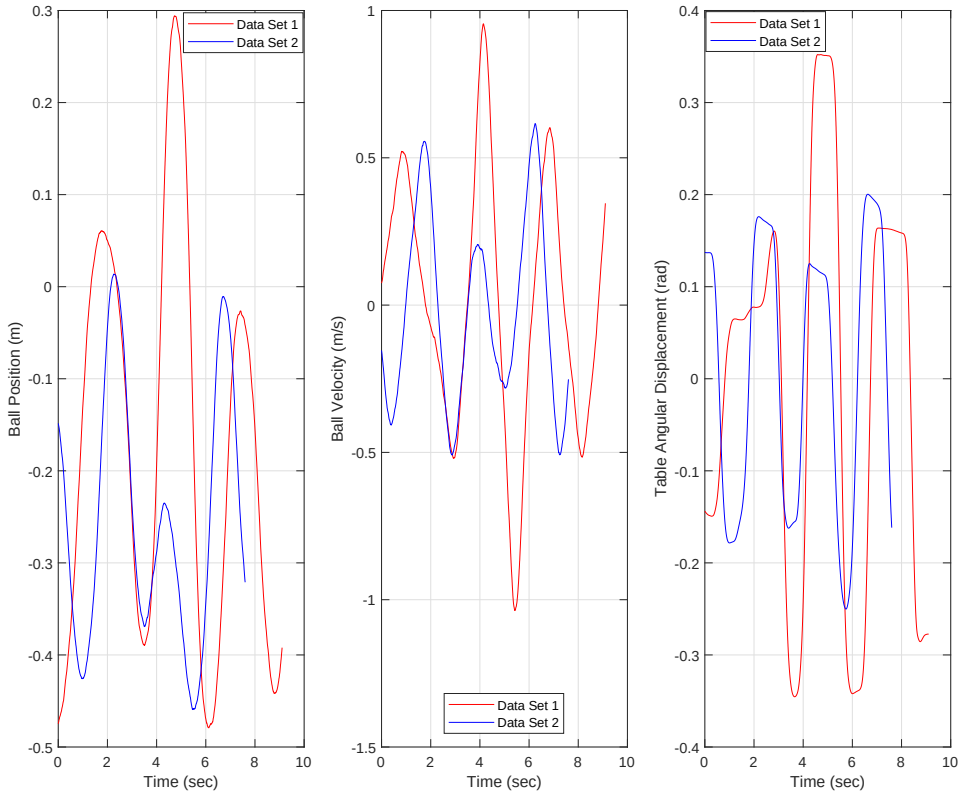

Figure 6.5: Plot of trajectories collected for Data Sets 1 and 2

r the radius of the ball used, l_h the horizontal distance from the pivot to the end of the table opposite the sensor, and x_s the measured reading from the distance sensor. Note that aside from the sensor reading, all parameters just defined are constant measurable of the experimental test bed. The position x in the $\hat{i}\hat{j}\hat{k}$ frame is given below in Equation (6.27).

$$x = l - r - l_h - x_s \tag{6.27}$$

Figures 6.6-6.8 demonstrate the transformation and smoothing performed on Data Set 1 as an example to provide insight into the noise characteristics of the system.

In Figure 6.6, it is clear that there is noise in the data, particularly in the case of the laser range finder. Thus, a non causal moving average smoother with a windowsize of 105 was

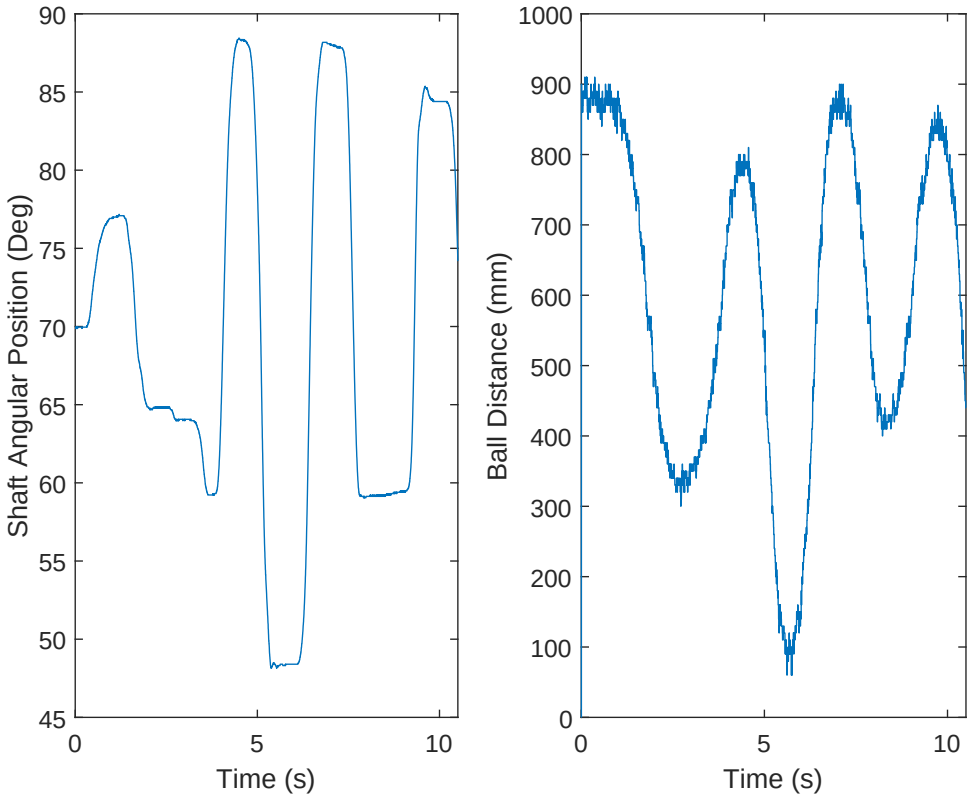

Figure 6.6: Plot of raw trajectories captured by system sensors

used to remove noise from the data after transformation to the proper frame. Figure 6.7 demonstrates the smoothed, transformed data as compared to the raw, transformed data.

Now that the trajectories have been transformed to the appropriate coordinate system and smoothed, the derivatives of the ball position and table angular displacement can be calculated. To do this, a centered finite difference was used to approximate the derivative as found in [77] and given in Equation (6.28). Note that Equation (6.28) does not actually use the function values at a given time t, but the symmetric nature about the point provides a more accurate approximation to the derivative when the time step T is small.

$$f'(x) = \frac{f(x+T) - f(x-T)}{2T} \tag{6.28}$$

As numerical differentiation induces high frequency noise, moving average filters were

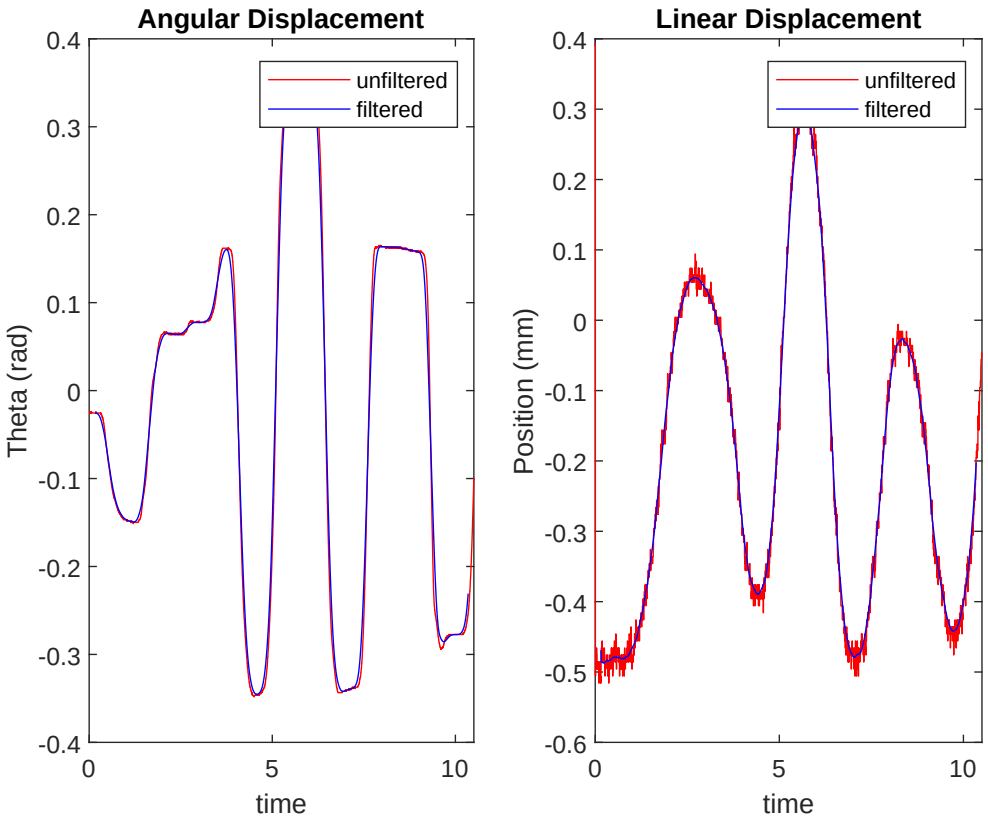

Figure 6.7: Plot comparing raw and smoothed, transformed system collected trajectories

applied to $\dot{\theta}$, $\dot{x}$, and $\ddot{\theta}$ of windowsizes 35, 125, 105 respectively in order to generate accurate representations of the true values. The derivatives of the state and control input trajectories are given in Figure 6.8. Note that in Figure 6.8, the trajectory corresponding to $\ddot{x}$ is unused as it does not appear in the discretized equations of motion anywhere; it is provided here for completeness and to demonstrate the high frequency noise introduced through numerical differentiation.

6.4.3 Experimental Procedure

In order to explore the effectiveness of the CEM on system identification tasks utilizing experimentally collected data, a series of model fitting studies were run with their procedure defined here. In order to quantify the effectiveness of a model, testing it over unseen

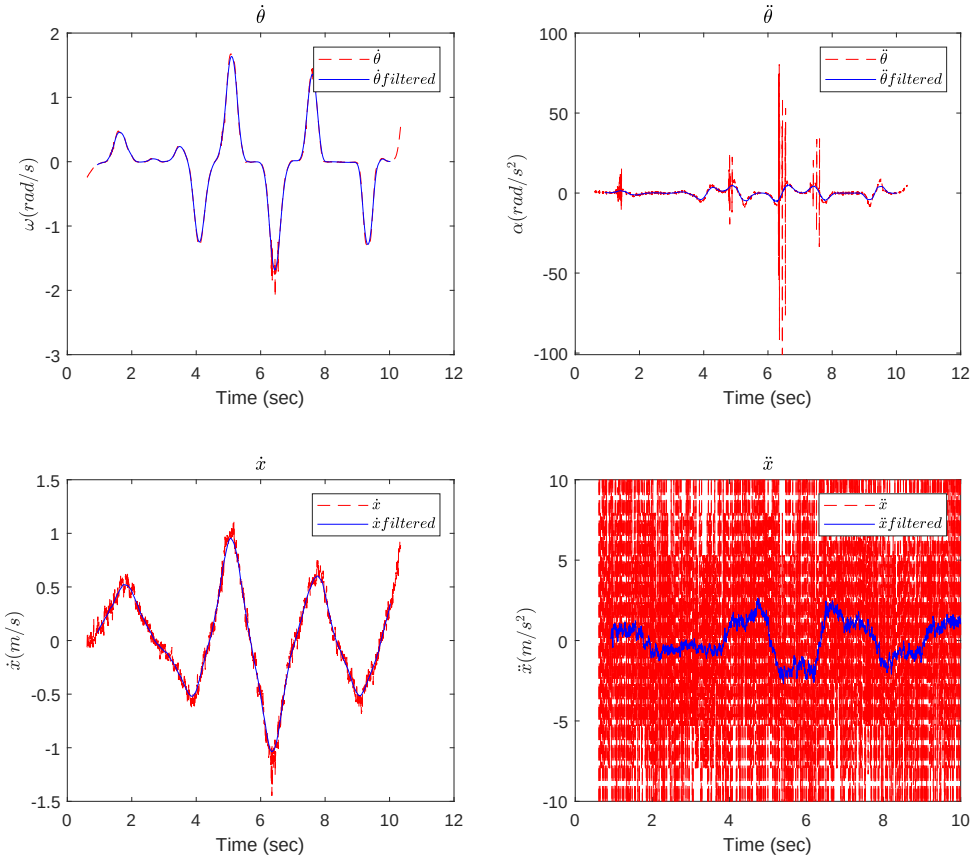

Figure 6.8: State and control input trajectories' derivatives

data allows for identification of overfit parameter sets. To this end, for the remainder of this work, Data Set 1 was arbitrarily selected as the training data set and Data Set 2 the validation data set. The term training set is used to signify that the trajectory was used to train the prospective model through output error minimization between the collected Data Set 2 trajectory and the output of a propagated proposed parameter set [8]. Data Set 2 is termed the validation set as it is unseen during parameter optimization and is instead used to quantify model performance by comparing the results of the collected data in Data Set 1 to the forward propagated model when provided with corresponding initial conditions and control inputs.

The experimental procedure was held constant and repeated for a variety of models,

which will be thoroughly described when introduced. The procedure for each proceeds as follows. When a prescribed model of a given structure is to be evaluated, output error minimization is run with Data Set 1 (the training set) using initial parameter guesses drawn from a zero-mean normal distribution with a standard deviation of 0.25. The optimized model is then propagated using the initial conditions and control inputs from Data Set 2 with the mean squared error between the true and predicted trajectories computed. In order to allow for study if the model tends to converge to various different local extrema, the above procedure was repeated for 100 different sets of initial parameter guesses with the optimization and propagation performed and averaged across all sets. This procedure is nearly identical to that used for a multi-start procedure frequently used for global optimization [78, 79]. Results of the average MSE over the 100 iterations are reported along with a plot of a trajectory from a singular iteration to provide some insight into the meaning of the absolute error value reported. Note that for each of the 100 iterations, a separate model is trained, propagated and the its propagation error computed; the name Model 1, 2, etc. is referring to the class or structure of the model used not a specific set of optimized parameters.

Throughout Sections 6.5 and 6.6, optimizations are run on various different models; each will be detailed here and summarized in Table 6.1. Further explanation will be given for each upon their first appearance. The simplest model, referred to throughout this work is Model 1, which contains no knowledge of the underlying generative dynamics and thus considers every potential parameter within the Θ matrix as potentially nonzero. The total number of parameters to be optimized is governed by the size of the potential function vector $\mathbf{F}$ considered. Model 2 is a model informed by the idealized dynamics given in Equation (6.26). Only parameters that are nonzero in Equation (6.26) are included in the optimization of Model 2. Model 3 represents a model informed by the CEM structure identified when considering the potential function vector $\mathbf{F}$ from the proposed minimal dynamic representation of the system. Only parameters identified as nonzero by the corresponding CEM are included in the optimization. Finally, Model 4 considers the case of a model

Table 6.1: Summary of models used for model and CEM performance quantification

Model Name	Model Description
Model 1	All parameters included
Model 2	Parameters according to idealized parameters included
Model 3	Parameters according to CEM computed on minimal dynamics included
Model 4	Parameters according to CEM computed on expanded dynamics included

Table 6.2: Comparison of predictive performance of Models 1 and 2 on the unseen Data Set 2

Model	Mean Validation MSE	Validation MSE Std. Dev.	Min Validation MSE	Max Validation MSE
Model 1	8.19	63.11	0.0743	605.7
Model 2	0.0032	0.0249	2.93×10^{-05}	0.2390

informed by CEM estimation on an expanded **F** as will be discussed below.

6.5 Model Validation

6.5.1 Parameter Set Comparison and Performance

Equation (6.26) provides a discrete time model of the physical system. In order to validate Model 2, it was compared to the case of Model 1, which includes all of the potential functions included in the **F** from Equation (6.26). The results of the comparison are displayed in Table 6.2.

A plot of sample results of the forward propagated Models 1 and 2 for a given initial parameter guess set is provided in Figure 6.9.

Table 6.2 demonstrates that Model 2 provides a far lower mean MSE, which signifies trajectories generated by Model 2 were on average far more accurate at predicting the unseen trajectory than those generated by Model 1. Additionally, the lower average stan-

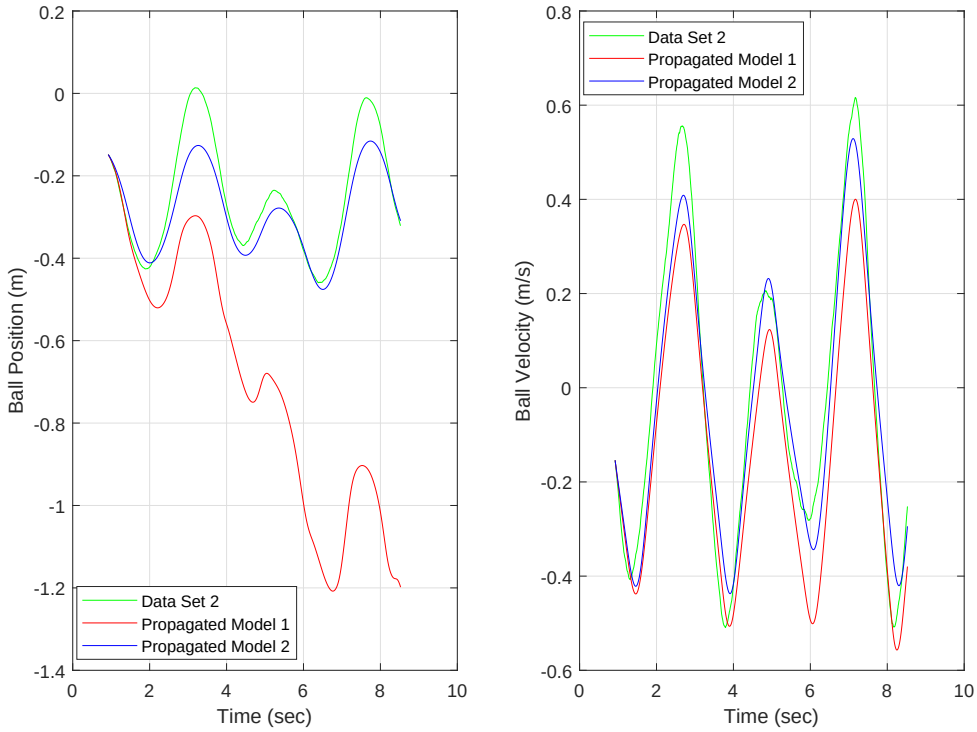

Figure 6.9: Forward propagated trajectories using validation initial conditions and control inputs for optimized Models 1 and 2 compared to the true collected state values

dard deviation suggests that there is less variation in models identified by Model 2, which means there is a decreased reliance on accurate initial conditions and and a lesser risk of convergence to a local minima, which will provide far worse results when used on data significantly different from that used for training. Figure 6.9 demonstrates sample trajectories of each and shows that Model 1 provides poor predictive accuracy specifically for the ball position, which is not surprising as the state equation for the ball position has a greater number of unneeded parameters (3) as opposed to the ball velocity (1). This example illustrates the importance of accurate covariate selection and model structure identification when it comes to optimization accuracy. This relatively simple example with only 10 parameters (of which only 4 are unneeded) can lead to large problems with convergence to a local extrema leading to poor model performance. The trajectory relating to Model 1 for

the ball position in Figure 6.9 significantly deviates from the true value contained in Data Set 2 and thus suggests Model 1 is a poor model for the data. Model 2's ability to converge to very similar models that all accurately allow for prediction of untrained data suggests that the structure of Model 2 does in fact generate a high fidelity model of the system.

6.6 CEM Computation Results

6.6.1 Minimal Potential Function Space

The CEM was computed using Data Set 1 when considering the minimal potential function set with the result given in Equation (6.29). Per the methodology discussed in Section **??**, the joint entropy of the states was computed for various durations of time series by considering time series starting at the first data point and each subsequent time series length increased by 25 data points. The results of the joint entropy with respect to the number of data points considered is given in Figure 6.10.

The initial major spike in the joint entropy is due to the PDF being poorly estimated and thus leading to a poor estimation of the joint entropy, so the values were ignored. The maximum joint entropy occurs at 2001 data points included with a maximum joint entropy of 13.84 nats. Thus, the first 2001 data points were included in computation of the CEM, which is given below in Equation (6.29).

$$CEM = \begin{bmatrix} 5.12 & 1.04 & 0.00 & 0.00 & 0.00 \\ 0.00 & 5.43 & 0.00 & 0.035 & 0.137 \end{bmatrix} \tag{6.29}$$

Models using the structure identified in Equation (6.29) will be referred to as Model 3. In Equation (6.29), the CEM achieved a covariate selection accuracy of 90% with the only incorrect entry occurring in the $(2, 3)$ entry. In order to test the performance of the CEM, an experiment identical to that used for model validation was run except Model 3 (the CEM informed model) was also included in the study with the results provided in Table 6.3. The experiment was again run for 100 iterations with the results averaged. The initial guesses

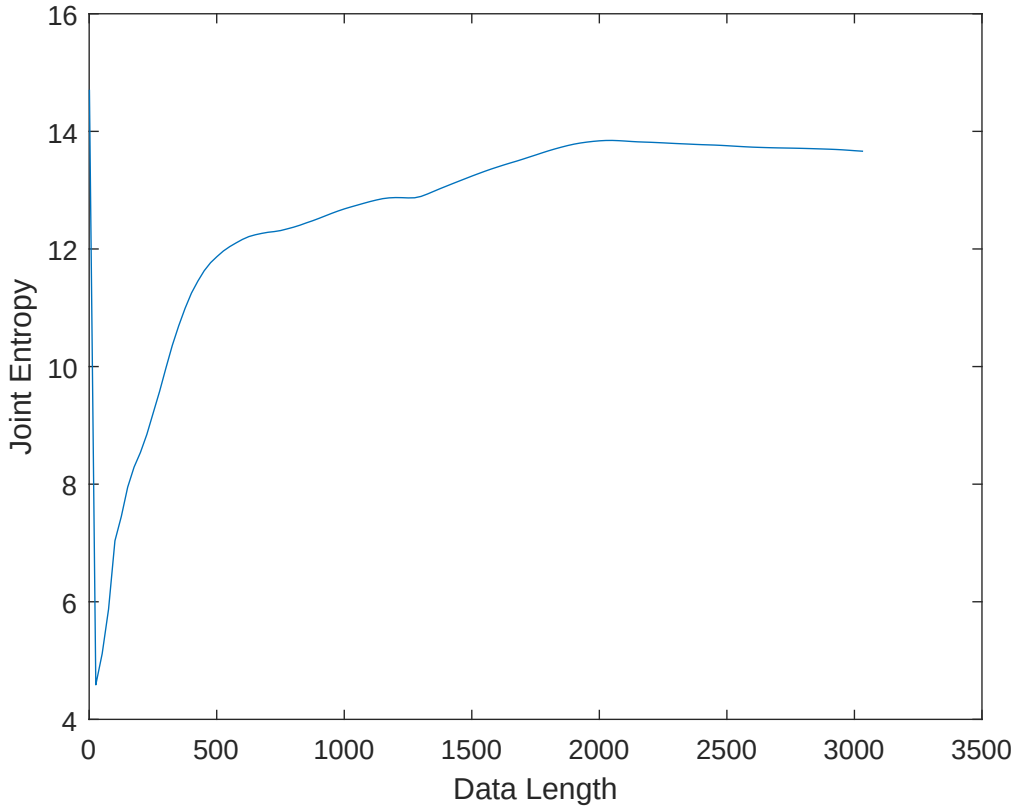

Figure 6.10: Joint entropy of the states as a function of the number of data points included in the estimation of the joint entropy

for the parameters were drawn from an identical distribution as previously used.

Considering Table 6.3, it is clear that Models 2 and 3 have nearly identical predictive performance. The difference in error metrics are small as the values of all criteria presented are on the same order of magnitude. Additionally, both far outperform the performance of Model 1 in every metric considered. Model 1, the full parameter set, was largely unable to correctly identify an adequately sparse model and converged instead to local extrema,

Table 6.3: Comparison of predictive performance of Models 1, 2 and 3

Model	Mean Validation MSE	Validation MSE Std. Dev.	Min Validation MSE	Max Validation MSE
Model 1	15.96	117.4624	0.0726	1066.4
Model 2	0.0443	0.0658	0.0111	0.1752
Model 3	0.0316	0.0547	0.0099	0.1770

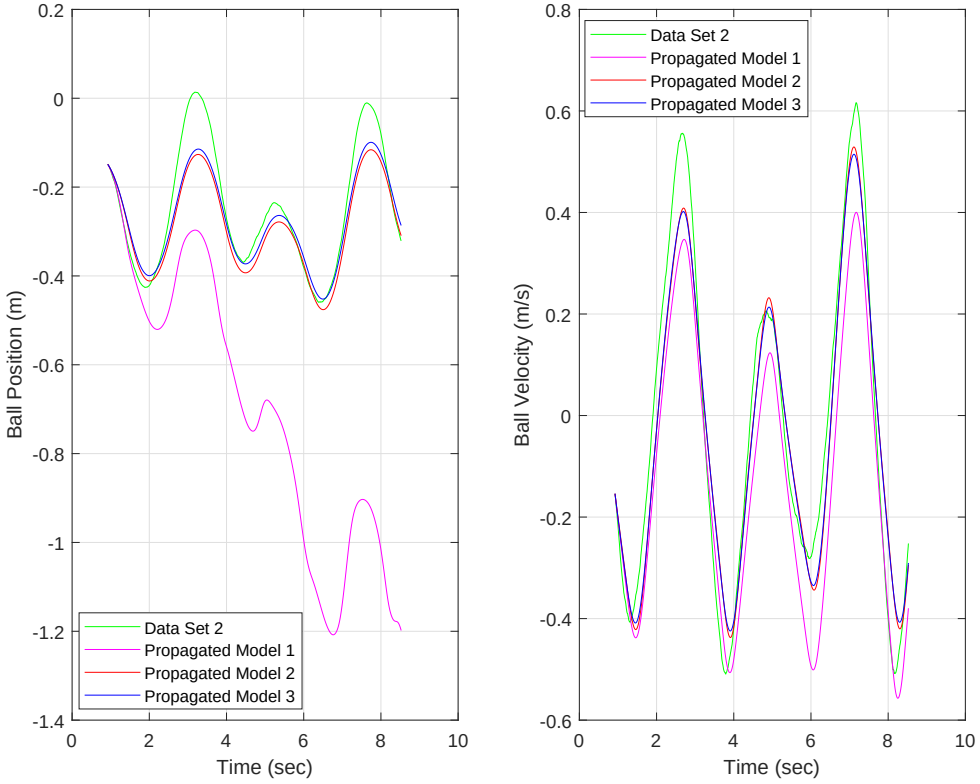

Figure 6.11: Example optimized trajectories using Models 2 and 3 compared with Data Set 2

which provided significantly degraded performance on average, and based on the maximum validation MSE, could potentially lead to an unstable model. Figure 6.11 demonstrates samples of the propagated model trajectories from a given set parameter initial guesses for visualization of sample results in Table 6.3. These results demonstrate that Models 2 and 3 have nearly identical predictive accuracy and both provide equally appropriate models, with Model 3 containing fewer parameters.

6.6.2 Expanded Potential Function Space

The experiment above in Section 6.6.1 considered the optimization problem where only the minimal potential function space was considered. Even in this scenario, the uninformed

optimization problem proved to be problematic with only the 10 parameters. Here, an expanded parameter set is considered. The potential function set was expanded to include 2 additional potential functions that have no real impact on the idealized system dynamics to make the problem more similar to a black box problem where potentially wholly unneeded parameters are included. The potential function vector **F** was augmented with two potential functions as shown in Equation (6.30).

$$\mathbf{F} = \begin{bmatrix} x_1 & x_2 & \ddot{\theta} & \dot{\theta}^2 x_1 & \sin\theta & x_2^2 & \dot{\theta}^2 \end{bmatrix}^T \tag{6.30}$$

Based on this **F**, the CEM will now be 2×7 instead of 2×5, with the two new entries at the end of each row ideally equal to 0. The CEM was again computed for the new system with the results given in Equation (6.31).

$$CEM = \begin{bmatrix} 5.20 & 1.08 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 5.47 & 0.00 & 0.00 & 0.164 & 0.00 & 0.00 \end{bmatrix} \tag{6.31}$$

Models with the structure of the CEM in Equation (6.31) will be referred to as Model 4. The CEM has a covariate selection accuracy of 85.7% as it again fails to identify the need for the $(2,3)$ parameter corresponding to the angular acceleration $\ddot{\theta}$ as in Equation (6.29); however, in this case the $(2,4)$ parameter corresponding to $\dot{\theta}^2 x_1$ was also incorrectly identified as zero when it was correctly identified in Equation (6.29). Notice that in Equation (6.29), the $(2,4)$ entry had the smallest causation entropy magnitude, which signifies that the CEM had the least certainty about said parameter and suggests it has a lowe sensitivity. The addition of the new potential functions complicates the underlying KDE problem and leads to the loss of the (2,3) parameter.

Section 5.2.2 demonstrated the effects of increased problem dimension on the accuracy of CEM estimation in the presence of noise. Additionally, in Section 5.1.2 it was demonstrated that in the presence of noise, parameters with the lowest sensitivities are lost

Table 6.4: Comparison of predictive performance of Models 2 and 4

Model	Mean Validation MSE	Validation MSE Std. Dev.	Min Validation MSE	Max Validation MSE
Model 2	0.0426	0.064	0.011	0.1752
Model 4	0.0361	0.057	0.013	0.1769

first due to measurement noise, which muddles the causal relationships between random variables driving the causation entropy to zero. In this case where there is both Gaussian measurement noise as well as potentially nonzero mean bias due to the real sensors, the parameter identified as the least importance or with the least sensitivity was lost. The increase of the dimension of the CEM estimation problem the CEM covariate selection capabilities decreased with the lowest sensitivity parameter lost.

In order to quantify the degradation, a comparison of the propagation results was again run this time using Model 4 and Model 2 (the parameter set informed by the minimal function set CEM). An experimental setup similar to the one previously used was run with 100 iterations considered. Initial guesses for the parameter values were drawn from the same distribution as previously used. The results are given in Table 6.4 with trajectories in Figure 6.12. Both Models 2 and 4 provide nearly identical performance. Both Models fit the unseen Data Set 2 and provide relatively similar trajectories approximating said data. This means that the lost parameter had a very low parameter sensitivity as differences in the parameter value (i.e. zero verse nonzero) had minimal effect on the model output. This not only provides more evidence to the previously demonstrated results on parameter loss due to noise, but also shows that even in the presence of various types of noise, the CEM was able to find an accurate, reduced order model that maintained high predictive accuracy that matched that of the idealized dynamics.

6.6.3 CEM Interpretation

When considering the models returned by the CEM in Equations (6.29) and (6.31), the entries related to angular acceleration $\ddot{\theta}$ and the centripetal acceleration $\dot{\theta}^2$ both had very

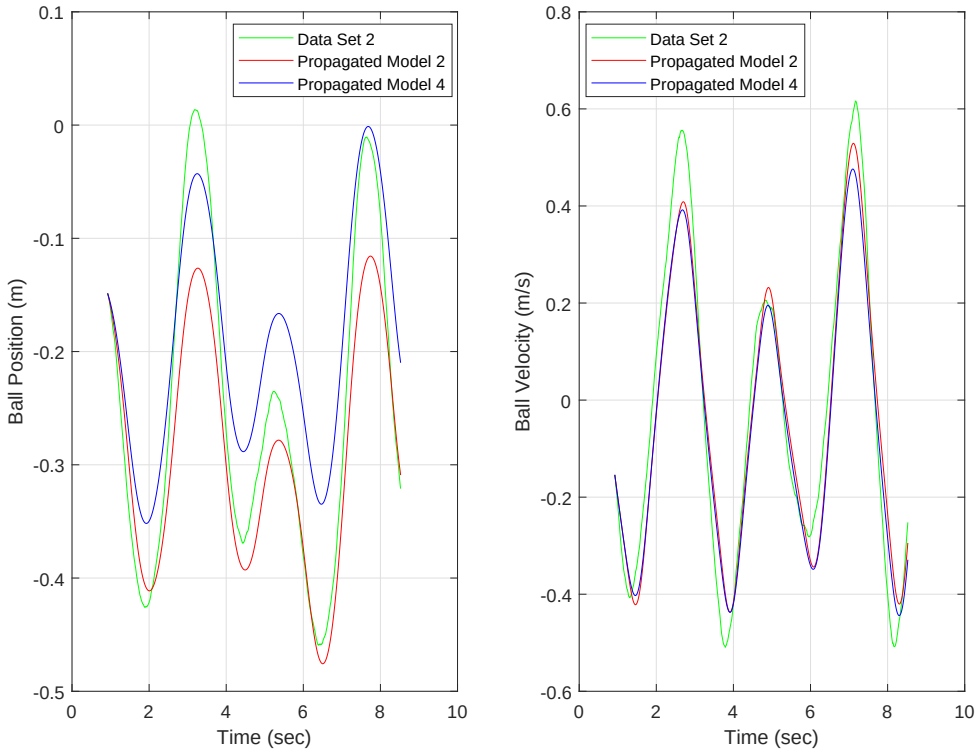

Figure 6.12: Example optimized trajectories using Models 2 and 4 compared with Data Set 2

low or incorrectly identified zero causation entropies. Part of the cause of the incorrect identification is the fact that there is noise in the collected measurement data and further noise introduced through the differentiation. However, if it was purely noise degrading the performance of the CEM, one would expect for the propagation MSE on the unseen Data Set 2 to be very large as there will be insufficient functions contained in the model to completely characterize the trajectory. However, the MSE was relatively low with the reduced parameter models (Models 3 and 4) still able to accurately predict system behavior. This suggests instead that the low or zero causation entropy values encountered with the parameters corresponding to $\ddot{\theta}$ and $\dot{\theta}x_1^2$ suggests that these functions actually have a low impact on the actual state trajectories.

The model returned in Equation (6.31) is essentially a quasi-static representation of the

system; it is the discrete model that represents a ball rolling down a slope with constant angle. The other acceleration terms have less of an impact on the systems studied, likely somewhat due to limitations on the physical system. Due to the length restriction of the actual system and the requirement that ball roll without slipping, exciting the modes relating to the angular and centripetal accelerations is difficult, which explains the CEM's difficulty in identifying the importance of the parameters. Similarly, the effects on the overall dynamics are relatively small and largely dominated by the angle of the table. Thus, even when the terms are lost from the CEM, the corresponding optimized model is still able to predict system behavior with a negligible change in predictive performance as all dominant components of the system are included in the system model.

The results above demonstrate the applicability of the CEM to the covariate selection problem for mechanical systems. This experimental example demonstrates that the CEM identifies the necessary parameters to limit model overfitting while still maintaining an accurate representation of the data whether seen or unseen during training. In this example, the idealized dynamics were derived, but it turns out for the system in question, the full dynamics were unnecessary to predict the system behavior, and a simpler model sufficed. With no apriori knowledge, the CEM identified that the dynamics were such that the angular and centripetal accelerations need not be considered. This is a positive result for the user as the higher the dimension an optimization problem, the higher the risk of overfitting or having issues of converging to the ideal, global solution. Reduction of the dimension of the optimization problem without any consequent reduction in the accuracy of the model fit will certainly benefit a user attempting to perform system identification on a nonlinear system. Thus, the CEM presents as a strong potential tool to aid in offline covariate selection tasks for mechanical systems.

CHAPTER 7
CONCLUSION

7.1 Research Summary

This work explores the applicability of the recently proposed causation entropy (and associated Causation Entropy Matrix) for covariate selection and model structure identification. Identification of model structure allows for improved optimization results through pre-optimization removal of unnecessary parameters to reduce both the order of the problem and thus the chances of parameter overfitting or convergence to a local extrema.

The causation entropy was proposed as a metric that identifies unique, causal information flow between random variables. Previous works proposed the application of the causation entropy to time series generated by mechanical systems by treating the state time histories as random variables. Previous work considered restricted, simple mathematical oscillators that contained data perfectly generated and sampled by the governing dynamics. This work expands upon the previously defined CEM by expanding the definition to allow for application to a very broad band of nonlinear dynamic systems. The Causation Entropy Matrix has an identical sparsity structure to that of the governing dynamics. This is demonstrated by the first demonstration of the application of the CEM to complex mechanical systems. It is also demonstrated that the magnitude of the nonzero entries in the CEM provides an estimate of the corresponding true model's relative parameter sensitvity, which can be useful for model order reduction as well as knowledge on where to focus research to best improve optimization results.

This work then explored the application of the CEM to black box systems and compared the optimized model performance to that of the cutting edge techniques of LASSO and Elastic Net. The CEM provides greater covariate selection accuracy and thus improved

model propagation performance in the presence of no measurement noise or mismatch. In the presence of measurement noise, it is demonstrated that the CEM will provide a sparser model than that of traditional shrinkage techniques, which includes a set of benefits and trade-offs discussed in the relevant section.

After the exploration of applications for the CEM, this work provides insights and techniques necessary for users to successfully use the CEM on various systems. This includes a first study of the effects of various types of noise (including Gaussian measurement noise and model mismatch/unmodeled dynamics) on causation entropy estimation. This includes a theoretical discussion of the impacts of the various types of noise on the computed causation entropy values assuming perfect underlying PDF knowledge as well as an in depth discussion of the impacts of measurement noise on the PDF estimation and how it propagates through the causation entropy estimation. A proposed method is derived and validated with experimental results to inform the user on how to best select the amount of data to maximize the performance of the CEM. Inclusion of maximal available data is not necessarily optimal as portions of the data could include no new excitation/information yet still add noise to the PDF estimation problem and thus degrade performance.

The work concluded with a study of the application of the CEM to data collected from an actual physical system. A system comprised of a ball rolling along a controlled, pivoting table was performed with sensor data collected on the table's angular displacement as well as the linear position of the ball. The sensor data was transformed and used to calculate all necessary state data to completely describe the behavior of the system. Details are provided on the necessary filtering and data processing for CEM estimation. Results are then provided on the covariate selection accuracy of the CEM along with a comparison on propagated model performance of the optimized model with and without knowledge gained from model structure estimation based on the CEM.

7.1.1 Technique Significance and Limitations

This work has demonstrated the applicability of a new technique for covariate selection of nonlinear systems that are linear in their parameterization. This allows identification of a lower order model that can have significant benefits in terms of avoiding parameter overfitting and convergence to local extrema by decreasing the dimension of the space the optimization is occurring in. The technique is demonstrated for the first time on a wide class of discretized mechanical systems. The technique avoids the need for any sort of hyperparameters or cross validation schemes like many leading current techniques (such as LASSO and elastic net). The CEM has the ability to identify the underlying model structure with higher accuracy and with less available data than standard optimization or LASSO and elastic net techniques, especially when only small amounts of noise are present. Additionally, the relative magnitudes of the nonzero parameters allow for insights to the related parameter sensitivity without requiring testing over multiple time series or estimation of gradients in high dimension.

The proposed CEM methodology has a lot of potential for improving covariate selection and thus the related parameter optimization problem; however, the technique is not perfect and still has its own set of drawbacks and limitations as most methods do. First, computation of the CEM is numerically expensive, which precludes its usage for online identification tasks. Additionally, only systems that are linear in their parameterization can viably be used in conjunction with the CEM technique. Note that parameters that are multiplied are possible through considering the product as its own parameter and then examining the observability of the system, but the verification is left for future work. Additionally, CEM estimation requires PDF estimation in high dimension. This work considers only one proposed KDE technique, which is by no means exhuastive; however; high dimension PDF estimation is known to be a difficult problem that grows significantly more challenging the higher the dimension space considered. This leads to problems with large decreases in accuracy as the noise level experience in the data increases. The technique is

rather sensitive to both noise and the dimension of the problem considered, though it does behave in predictable ways for given sources of error as shown herein.

Thus, like most techniques the CEM has both advantages and disadvantages that must be weighed given a specific task at hand to determine if the technique is appropriate.

7.2 Potential Avenues for Future Work

The developments presented provide many new insights and techniques for the application of the CEM for system structure identification. Questions were answered about the applicability and performance of the CEM; however, with more understanding comes the potential for more questions. This section will outline some of the areas for potential future research.

7.2.1 Probability Density Estimation

This work has demonstrated the applicability of the CEM to problems of system identification for mechanical systems, though the results are obviously generalizable to other types of systems and applications. However, many of the results discussed center on issues with the KDE accuracy in estimating the underlying PDF. Thus, a first series of potential avenues to improve on the probability estimation problem are presented.

KDE Improvement

The kernel density estimator used in this work stems from an entropy estimator presented in [42] published in the mid-1990s. Since then there have been improvements in the area of Kernel Density Estimation that could potentially lead to improved density estimation results, a decreased computation load or both. First, estimators exist that do not rely on the covariance matrix or require its inversion (one of the most computationally expensive parts of the KDE method used) as it instead uses a diagonal matrix of one dimensional standard deviations. Such an estimator is presented in [43]; implementation of such an es-

timation technique would likely decrease the computational load of CEM estimation with the corresponding effect on CEM performance to be determined. Additionally, there has been developments in the area of covariance estimation by leveraging the structure to allow for improved accuracy [80]. Improved covariance estimation, particularly in higher dimensions, could potentially lead to improved PDF estimation and combat the issues generated by the curse of dimensionality. Finally, Scott has proposed work of reducing high dimensional KDE problems into lower dimensional problems through projection using principal components or projection pursuit [81]. Reduction of the dimension of the problem will certainly decrease the computational load and potentially increase accuracy as problems with empty spaces in the PDF can potentially be resolved.

KNN Estimator

The exploration of a KNN estimator proposed in [41] is discussed in Section 2.2.4. However, the results in Section 2.2.4, the KNN estimator was unable to correctly estimate entropies for more complicated nonlinear system. A discussion of the potential causes of failure is included in Section 2.2.4; however, there is still great potential in using a KNN estimator as the estimator was shown to be accurate for a simple example and requires a far smaller computational load than that required for KDE. Data normalization may prove to solve concerns with the KNN estimator. Simple data normalization may suffice [82], or recently more robust normalization techniques have been recommended [83]. Exploring the applicability and performance of a KNN estimator is an interesting source of research as the KNN estimator does not require exact PMF values and thus may perhaps have improved performance in the case of noise of various types.

Sieve Estimator

Recently, there has been work suggesting a newer kind of estimator that can improve upon entropy estimation. In [84], a new estimation technique called a sieve estimator that uti-

lizes Grenander's method of sieves for PDF estimation. It makes no assumptions on the underlying distribution (such as using a Gaussian kernel in KDE) to estimate the PDF by using increasingly more complicated approximations to the PDF as data becomes available [85, 86]. [84] provides an algorithm for a sieve estimator for Shannon entropy, which can be used to estimate the causation entropy. There has been limited testing reported on it for problems of this sort.

7.2.2 Model Complexity and Data Used

The above suggestions all centered around potential avenues for improving upon the probability estimation problem to yield improved performance from the CEM. This work assumed that processes are strictly Markovian; additionally a first order approximation to the derivative was used to discretize the system. Thus, only data immediately preceding a data point was considered; however, this leaves it potentially susceptible to noise and model mismatch. However, using a higher order derivative approximation and relaxing the requirement that $\tau = 1$ and consider a greater memory for the causation entropy matrix to yield improved results. In a similar vein, the Directed Information has been proposed [32, 34], which uses the causation entropy considering all subsets of time available from $t = t_f$ working back to t_0. In [32], the Directed Information was used to a similar end as the causation entropy, however, there has been no work done on estimation of the Directed Information for systems of the sort studied herein.

Finally, more advanced techniques for data filtering could be considered to attempt to handle noise present in a system and achieve the most accurate representation of the data before computing the CEM. This work considered only a non-causal moving average filter. However, there are a wide range of available filters, that could be applied to attempt to improve the data and thus create a corresponding improvement in the estimated CEM. It was demonstrated in this work that the inclusion of appropriately tuned filters or smoothers can have a positive impact on the performance of the CEM; thus, it stands to reason that further

improved filtering techniques may create an even greater improvement in the performance of the CEM.

REFERENCES

[1] J.-N. Juang and R. S. Pappa, "An eigensystem realization algorithm for modal parameter identification and model reduction," *Journal of Guidance, Control, and Dynamics*, vol. 8, no. 5, pp. 620–627, 1985.

[2] J.-N. Juang, M. Phan, L. G. Horta, and R. W. Longman, "Identification of observer/kalman filter markov parameters: Theory and experiments," *Journal of Guidance, Controls, and Dynamics*, vol. 16, no. 2, pp. 320–329, March-April 1993.

[3] L. Ljung, "System identification," in *Signal Analysis and Prediction. Applied and Numerical Harmonic Analysis*, A. Prochazka, J Uhlir, P. Rayner, and N. Kingsbury, Eds., Boston, MA: Birkhauser, 1998, pp. 163–173.

[4] K. J. Keesman, *System Identification*. Springer-Verlag London, 2011.

[5] O. Nelles, *Nonlinear System Identification: From Classical Approaches to Neural Networks and Fuzzy Models*. Springer Science and Business Media, 2013.

[6] J.-N. Juang, *Applied System Identification*. Upper Saddle River, NJ: Prentice Hall, 1994.

[7] D. G. Luenberger and Y. Ye, *Linear and Nonlinear Programming Fourth Edition*. Springer, 2016.

[8] P Kabaila, "On output-error methods for system identification," *IEEE Transactions on Automatic Control*, vol. 28, no. 1, pp. 12–13, 1983.

[9] K Iliff, "Parameter estimation for flight vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 12, no. 5, pp. 609–622, 1989.

[10] B. Taylor and J. Rogers, "Experimental investigation of real-time helicopter weight estimation," *Journal of Aircraft*, vol. 51, no. 3, pp. 1047–1051, 2014.

[11] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, "Identifying and attacking the saddle point problem in high-dimensional non-convex optimization," in *Advances in neural information processing systems*, 2014, pp. 2933–2941.

[12] L. Perea, J. How, L. Breger, and P. Elosegui, "Nonlinearity in sensor fusion: Divergence issues in ekf, modified truncated gsf, and ukf," in *Gguidance, Navigation and Control Conference and Exhibit*, AIAA, 2007.

[13] A. Gelb, *Applied Optimal Estimation*. MIT Press, 1974.

[14] B. Ristic, S. Arulampalam, and N. Gordon, *Beyond the Kalman filter particle filters for tracking applications*. Boston, MA.: Artech House, 2004.

[15] D. M. Hawkins, "The problem of overfitting," *Journal of Chemical Information and Computer Science*, vol. 44, no. 1, pp. 1–12, 2004.

[16] G. Kerschen, K. Worden, A. F. Vakakis, and J.-C. Golinval, "Past, present and future of nonlinear system identification in structural dynamics," *Mechanical Systems and Signal Processing*, vol. 20, no. 3, pp. 505–592, 2006.

[17] P. Kim, J. Rogers, J. Sun, and E. Bollt, "Causation entropy identifies sparsity structure for parameter estimation of dynamic systems," *Journal of Computational and Nonlinear Dynamics*, vol. 12, no. 1, 2017.

[18] J Sun and E. Bollt, "Causation entropy identifies indirect influences, dominance of neighbors and anticipatory couplings," *Physica D*, vol. 267, pp. 49–57, 2014.

[19] J. Witte and V. Didelez, "Covariate selection strategies for causal inference: Classification and comparison," *Biometrical Journal*, vol. 61, no. 5, pp. 1270–1289,

[20] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 2nd ed. Pearson Education, 2003, ISBN: 0137903952.

[21] S. Maldonado and R. Weber, "A wrapper method for feature selection using support vector machines," *Information Sciences*, vol. 179, no. 13, pp. 2208–2217, 2009.

[22] D. Bertsimas, A. King, and R. Mazumder, "Best subset selection via a modern optimization lens," *The annals of statistics*, pp. 813–852, 2016.

[23] S. Das, "Filters, wrappers and a boosting-based hybrid for feature selection," in *Icml*, vol. 1, 2001, pp. 74–81.

[24] M. A. Babyak, "What you see may not be what you get: A brief, nontechnical introduction to overfitting in regression type models," *Psychosomatic Medicine*, no. 3, pp. 411–421, June 2004.

[25] H. Zou, T. Hastie, and R. Tibshirani, "On the degrees of freedom of the lasso," *The Annals of Statistics*, vol. 35, no. 5, pp. 2173–2192, 2007.

[26] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B*, vol. 58, pp. 267–288, 1 1996.

[27] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society Statistical Methodology Series B*, vol. 67, no. 2, pp. 301–320, 2004.

[28] R. J. Tibshirani, "The lasso problem and uniqueness," *Electronic Journal of Statistics*, vol. 7, 2013.

[29] P. Zhao and B. Yu, "On model selection consistency of lasso," *Journal of Machine Learning Research*, no. 7, 2006.

[30] L. Barnett, A. B. Barrett, and A. K. Seth, "Granger causality and transfer entropy are equivalent for gaussian variables," *Physical review letters*, vol. 103, no. 23, p. 238 701, 2009.

[31] R. Vicente, M. Wibral, M. Lindner, and G. Pipa, "Transfer entropy—a model-free measure of effective connectivity for the neurosciences," *Journal of computational neuroscience*, vol. 30, no. 1, pp. 45–67, 2011.

[32] J. Massey, "Causality, feedback and directed information," in *Proc. Int. Symp. Inf. Theory Applic.(ISITA-90)*, Citeseer, 1990, pp. 303–305.

[33] G. Kramer, *Directed information for channels with feedback*. Citeseer, 1998.

[34] C. J. Quinn, N. Kiyavash, and T. P. Coleman, "Directed information graphs," *CoRR*, 2012.

[35] C. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, 379–423 and 623–656, July 1948.

[36] T Cover and J Thomas, *Elements of Information Theory*. New York: John Wiley and Sons, 1991.

[37] T Schreiber, "Measuring information transfer," *Physics Review Letters*, vol. 85, pp. 461–464, 2000.

[38] J. Sun, D. Taylor, and E. M. Bollt, "Causal network inference by optimal causation entropy," *SIAM Journal on Applied Dynamical Systems*, vol. 14, no. 1, pp. 73–106, 2015.

[39] J. Sun and E. Bollt, "Causation entropy identifies indirect influences, dominance of neighbors and anticipatory couplings," *Physica D*, vol. 267, pp. 49–57, 2014.

[40] J. Sun, C. Cafaro, and E. Bollt, "Identifying the coupling structure in complex systems through the optimal causation entropy principle," *Entropy*, vol. 16, no. 6, pp. 3416–3433, 2014.

[41] A. Kraskov, A. Stogbauer, and P. Grassberger, "Estimating mutual information," *Physical Review E*, vol. 69, 2004.

[42] Y.-I. Moon and B. Rajagopalan, "Estimation of mutual information using kernel density estimators," *Physical Review E*, vol. 52, no. 3, pp. 2318–2321, 1995.

[43] D. W. Scott, *Multivariate Density Estimation Theory, Practice and Visualization*. John Wiley and Sons, 1992.

[44] B Silverman, *Density Estimation for Statistics and Data Analysis*. New York, NY: Springer Science and Business Media, 1986.

[45] J. Beirlant, E. Dudewicz, L. Gyor, and E. Meulen, "Nonparametric entropy estimation: An overview," *International Journal of Mathematical and Statistical Sciences*, vol. 6, 1997.

[46] E. Archer, I. M. Park, and J. W. Pillow, "Bayesian entropy estimation for countable discrete distributions," *Journal of Machine Learning Research*, vol. 15, no. 1, October 2014.

[47] C. M. Grinstead and J. L. Snell, *Introduction to Probability*. American Mathematical Society, 2012.

[48] D. V. Griffiths and I. Smith, *Numerical Methods for Engineers*, 2nd ed. CRC Press, 2006.

[49] K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[50] J. H. Friedman, R. Tibshirani, and T. Hastie, *The Elements of Statistical Learning*. Springer, 2001.

[51] N. Ahmed and D. Gokhale, "Entropy expressions and their estimators for multivariate distributions," *Transactions on Information Theory*, vol. 35, 3 May 1989.

[52] K. Beyer, "When is nearest neighbor meaningful," in *International conference on database theory*, Berlin, Heidelberg: springer, 1999.

[53] W. M. Lord, J. Sun, and E. M. Bollt, "Geometric k-nearest neighbor estimation of entropy and mutual information," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 28, no. 3, 2018.

[54] A Barrat, M Barthelemy, and A Vespignani, *Dynamical Processes on Complex Networks*. Cambridge, England: Cambridge University Press, 2008.

[55] J Sun, C Cafaro, and E Bollt, "Identifying the coupling structure in complex systems through the optimal causation entropy principle," *Entropy*, vol. 16, pp. 3416–3433, June 2014.

[56] J. Elinger and J. Rogers, "Information theoretic causality measures for system identification of mechanical systems," *Journal of Computational and Nonlinear Dynamics*, vol. 13, no. 7, 2018.

[57] H. K. Khalil, *Nonlinear Control*. Prentice Hall, 2015.

[58] K. S. Narendra and A. M. Annaswamy, *International Journal of Control*, 1986.

[59] R. L. McCoy, *Modern Exterior Ballistics*. Schiffer Publishing, 2012.

[60] M. Gross, J. Rogers, and M. Costello, "Nonlinear stability analysis methods for guided artillery projectiles," *AIAA Atmospheric Flight Mechanics Conference*, June 2014.

[61] F. Fresconi, B. Guidos, I. Celmins, and W. Hathaway, "Flight behavior of an asymmetric body through spark range experiments using roll-yaw resonance for yaw enhancement," *AIAA Atmospheric Flight Mechanics Conference*, vol. 15, no. 1, October 2014.

[62] D. M. Hamby, "A review of techniques for parameter sensitivity analysis of environmental models," *Environmental Monitoring and Assessment*, vol. 32, no. 2, pp. 135–154, September 1994.

[63] R. Gardner, D. D. Huff, R. V. O'Neill, J. Mankin, J. Carney, and J. Jones, "Application of error analysis to a marsh hydrology model," *Water Resources Research*, vol. 16, no. 4, pp. 659–664, August 1980.

[64] R. V. O'Neill, R. Gardner, and J. Mankin, "Analysis of parameter error in a nonlinear model," *Ecological Modelling*, vol. 8, pp. 297–311, January 1980.

[65] X.-R. Cao, "Convergence of parameter sensitivity estimates in a stochastic experiment," *IEEE Transactions on Automatic Control*, vol. 30, no. 9, 1985.

[66] H. B. Nielsen and K. Madsen, *Introduction to Optimization and Data Fitting*. Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2010.

[67] K. H. Rosen, *Discrete Mathematics and Its Applications*. McGraw-Hill Higher Education, 2002.

[68] A. E.-N. S. Ahmed, A. S. Ali, N. M. Ghazaly, and G. A. el Jaber, "Pid controller of active suspension system for a quarter car model," *International Journal of Advances in Engineering and Technology*, vol. 8, no. 6, pp. 899–909, Dec 2015.

[69] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[70] S. Wiggins, *Introduction to Applied Nonlinear Dynamical Systems and Chaos*, Second. Springer-Verlag, 2003.

[71] S. Roweis, "Levenberg-marquardt optimization," *Notes, University Of Toronto*, 1996.

[72] S. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1997.

[73] W. H. Press, W. T. Vetterling, P. P. Flannery, and S. Teukolsky, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 1992, ch. 13.5.

[74] D. W. Scott and S. R. Sain, *Multi-dimensional density estimation*, 2004.

[75] A. A. Borovkov, *Probability Theory*. Springer-Verlag London, 2013.

[76] J. Ginsberg, *Engineering Dynamics*. Cambridge University Press, 2008.

[77] R. J. LeVeque, *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. SIAM, 2007.

[78] R. Martí, "Multi-start methods," in *Handbook of metaheuristics*, Springer, 2003, pp. 355–368.

[79] Z. Ugray, L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Martí, "Scatter search and local nlp solvers: A multistart framework for global optimization," *INFORMS Journal on Computing*, vol. 19, no. 3, pp. 328–340, 2007.

[80] T. T. Cai, Z. Ren, H. H. Zhou, *et al.*, "Estimating structured high-dimensional covariance and precision matrices: Optimal rates and adaptive estimation," *Electronic Journal of Statistics*, vol. 10, no. 1, pp. 1–59, 2016.

[81] D. W. Scott and J. R. Thompson, "Probability density estimation in higher dimensions," in *Computer Science and Statistics: Proceedings of the fifteenth symposium on the interface*, North-Holland, Amsterdam, vol. 528, 1983, pp. 173–179.

[82] C. C. Aggarwal, *Data mining: the textbook*. Springer, 2015.

[83] A. Pandey and A. Jain, "Comparative analysis of knn algorithm using various normalization techniques," *International Journal of Computer Network and Information Security*, vol. 9, no. 11, p. 36, 2017.

[84] L. Paninski, "Estimation of entropy and mutual information," *Neural Computation*, vol. 15, no. 6, pp. 1191–1253, 2003.

[85] S. Geman and C.-R. Hwang, "Nonparametric maximum likelihood estimation by the method of sieves," *The Annals of Statistics*, pp. 401–414, 1982.

[86] X. Shen, "On methods of sieves and penalization," *The Annals of Statistics*, pp. 2555–2591, 1997.

VITA

Jared Elinger was born and raised in Saint Petersburg Florida. In May 2016 he graduated Magna Cum Laude with a Bachelors of Science in Mechanical Engineering from Rice University. While at Rice he was a member of the Mechatronics and Haptic Interfaces (MAHI) Lab under Dr. Marcia O'Malley where he researched robotic rehabilitation methods for traumatic brain injury patients with incomplete spinal cord injuries as well as helped develop a tool for student instruction on haptics and system dynamics. While at Rice he interned at both the NASA Johnson Space Center as well as Schlumberger. After graduating from Rice, Jared moved to Atlanta to begin his PhD studies under Dr. Jonathan Rogers in the Intelligent Robotics and Emergent Automation Lab where he worked on both system identification as well as multi-agent control problems. His research interests include dynamic systems and controls, system identification and learning.